Universiteit Leiden

Optimizing SubDisc Subgroup Quality: A Comparative Study of Search Methods Across Interval Sizes

P.G. van der Steeg (s3312259)

Supervisors:
Dr. A.J. Knobbe & Dr. F. Bariatti

Bachelor Thesis– Data Science and Artificial Intelligence

Leiden Institute of Advanced Computer Science (LIACS)
liacs.leidenuniv.nl

June 28, 2026

Abstract

In this thesis we investigate whether post-hoc optimisation of numeric condition values can improve the quality of subgroups found by the SubDisc subgroup discovery system. SubDisc uses beam search to explore the search space, which is inherently greedy and may result in numeric condition values that are not jointly optimal in the final subgroup. To address this, four post-hoc search methods are applied to refine these values within a bounded interval around the original beam search result: full brute force, interval brute force, cross search, and hill climbing. The methods are evaluated on three benchmark datasets: Adult, German Credit, and Pima Indians Diabetes, across ten interval sizes ranging from 5% to 50%. Weighted Relative Accuracy (WRAcc) is used as the quality measure throughout. Additionally, a redundancy filter is applied to remove subgroups that converge to the same configuration after optimisation. For evaluation, we compare the methods on three dimensions: mean WRAcc improvement across interval sizes, the effect of redundancy filtering on the resulting subgroup sets, and the trade-off between computational cost and quality improvement. The results show that all methods improve subgroup quality to varying degrees, with cross search offering the best balance between improvement and runtime, and datasets with a higher proportion of numeric features showing the greatest potential for post-hoc optimisation.

Contents

1 Introduction

Subgroup Discovery (SD) is the subfield of data exploration that aims to discover interpretable descriptions of subsets of the data that stand out with respect to a given target variable, from now on referred to as *subgroups* [1]. Identifying these subgroups provides insight across domains including healthcare, industry, and social media, by revealing which segments of a population behave differently with respect to a target of interest [2].

To extract these subgroups from data, SD systems must search through a large space of possible subgroup descriptions. The space of possible subgroups grows exponentially with the number of features and their possible values or labels, making exhaustive search infeasible for large or complex datasets [3]. To remain feasible, SD systems employ heuristic search strategies that efficiently explore promising regions of the search space without evaluating every candidate [3].

This heuristic search reduces computational complexity while still finding good subgroups, but it is not guaranteed to find the optimal result. By greedily focusing on promising regions, it may miss high-quality subgroups elsewhere in the search space. Finding a good balance between broadly exploring the search space and exploiting known promising areas is a core challenge in SD [3]. This challenge is not only present with choosing the conditions that define a subgroup, but also within the conditions themselves. The numeric values or categorical labels for the different conditions that define a subgroup may not be jointly optimal, as each value or label is chosen greedily without consideration of the other conditions that will later form the subgroup. This thesis will focus on this particular possible loss of subgroup quality.

SD systems use a quality measure to evaluate how interesting a found subgroup is. This maps a subgroup to a number reflecting how interesting it is [2]. The choice of quality measure depends on the discovery task and target type. In this thesis, Weighted Relative Accuracy (WRAcc) is used, as it rewards subgroups that are both large and have a target rate that deviates substantially from the dataset average [4].

For the SD experiments in this thesis, we use SubDisc, a data mining tool for discovering subgroups in data. We configured SubDisc to use beam search for exploring the search space, with WRAcc as the quality measure and numeric strategy best for handling numeric attributes [5].

Since beam search is greedy, the numeric values in a final subgroup may not be configured optimally. This thesis investigates whether applying post-hoc search methods (full brute force, interval brute force, cross search, and hill climbing) to refine the subgroups numeric condition values within a bounded search space can improve subgroup quality, and how these methods compare in terms of quality gained and computational cost. Since post-hoc optimization can cause subgroups with the same conditions to converge to the same numeric configuration, a redundancy filter is applied to remove duplicate subgroups, retaining only the highest quality configuration.

1.1 Research Question

The field, methods and potential for improvement outlined above motivate the following research question and sub-questions:

Research Question: How do the post-hoc optimisation methods full brute force, interval brute force, cross search, and hill climbing compare when applied to SubDisc results to improve subgroup quality?

Sub-questions:

- What is the effect of each method on WRAcc improvement across different interval sizes?
- How does redundancy filtering affect the subgroup sets produced by each method?
- How does the computational cost of each method relate to the quality improvement achieved?

1.2 Thesis Overview

This thesis is structured as follows. Chapter 2 provides the necessary background, further explaining and defining Subgroup Discovery, the WRAcc quality measure, and SubDisc. It then describes the possible numeric strategies, the three post-hoc search methods investigated in this thesis: (interval) brute force, cross search, and hill climbing, and concludes with related work. Chapter 3 describes the experimental setup, including the three datasets used (Adult, German, and Diabetes), the SubDisc search parameter configuration, the interval sizes explored, and the evaluation metrics. Chapter 4 presents the experimental results per dataset, reporting the SubDisc baseline and comparing the post-hoc methods in terms of WRAcc improvement, redundancy filtering, and runtime. Chapter 5 discusses the limitations and validity of the experimental approach, presents the conclusions by answering the research question, and Chapter 6 suggests directions for further research.

2 Background

2.1 Subgroup Discovery

Subgroup Discovery (SD) is a local, supervised, descriptive pattern mining paradigm [6]. Given a dataset and a user-specified target variable, SD aims to find subgroup descriptions; combinations of conditions on features, where each condition constrains a feature to a specific value, range or label [7]. These descriptions define subsets of the data whose target distribution deviates from that of the dataset as a whole, which we call subgroups [7]. Subgroups are found through a level-wise search, starting with single conditions on individual features and refining promising candidates by adding more conditions, guided by a quality measure that expresses how exceptional a subgroup is [7]. The goal is not to find a single global model, as in classification, but to return a ranked list of the top-k most interesting subgroups, each evaluated independently [6].

In the search space, features can be nominal or numeric. Nominal features take a fixed set of discrete values or labels, making them relatively straightforward to handle. Numeric features

however can take a very large number of distinct values, potentially giving them high cardinality and making exhaustive evaluation of all possible conditions infeasible [7]. In this thesis, the target variable is always binary nominal, meaning one value is designated as the positive class and all others as negative.

An important aspect of SD is the occurrence of redundant subgroups. A subgroup can be considered redundant if it does not substantially differ from another subgroup in its description, the subset it covers, or its exceptional model [3]. In this thesis, redundancy is defined based on subgroup cover: if two subgroups cover the same subset of the data, the subgroup that has a superset of conditions is considered redundant and removed.

2.2 WRAcc

In this thesis, we use Weighted Relative Accuracy (WRAcc) which is a well-known SD quality measure for datasets with one binary target attribute [3].

The formal definition for WRAcc is as follows:

$$WRAcc(C \leftarrow D) = \frac{|D|}{|N|}\left(\frac{|D \cap C|}{|D|} - \frac{|C|}{|N|}\right)$$

Where:

- N is the dataset
- D is the subgroup
- C is the target class
- $\frac{|D|}{|N|}$ is the rule generality (relative size of the subgroup)
- $\frac{|D \cap C|}{|D|}$ is the target rate within the subgroup
- $\frac{|C|}{|N|}$ is the default target rate in the dataset

WRAcc consists of two components: rule generality $\frac{|D|}{|N|}$, which measures the relative size of the subgroup, and relative accuracy $\left(\frac{|D \cap C|}{|D|} - \frac{|C|}{|N|}\right)$, which measures how much the target rate within the subgroup deviates from the default accuracy of the dataset [4]. A subgroup is only considered interesting if it improves upon this default accuracy. By using generality as a weight, WRAcc balances subgroup size and target distribution deviation; a subgroup must be both sufficiently large and show a meaningful deviation in target rate to achieve a high score [4]. WRAcc is zero when the subgroup target rate equals the dataset average, positive when the subgroup has a higher target rate than the dataset average, and negative when it has a lower target rate.

2.3 SubDisc

SubDisc is a data mining tool made with Java for discovering local patterns in data, developed at LIACS [5]. It features a generic Subgroup Discovery algorithm that can be configured in many ways, supporting multiple data types for both input and target attributes, including nominal, numeric, and binary. SubDisc supports a range of quality measures and numeric strategies, making it flexible for different SD tasks [5]. In this thesis, SubDisc is configured to use beam search as the search method, WRAcc as the quality measure, and numeric strategy best for handling numeric attributes. SubDisc was previously developed under the name Cortana [5].

2.4 Beam Search

Beam search is a heuristic search strategy commonly used in SD when no feasible exhaustive algorithm exists [7]. The search proceeds level-wise through the search space, where at the first level, candidate subgroups are built by imposing one condition on one feature and all such candidates are considered. Two key parameters bound the search: the search width w and the search depth d [7]. After the first level, only the w most promising candidates are retained to form the beam for the next level. These candidates are then refined by adding one more condition, and again the w best are selected for the following level. This process continues until all candidates at depth d have been considered [7].

Beam search is inherently greedy; once a candidate is discarded it is never reconsidered, and the numeric values chosen for conditions are not revised as the subgroup grows deeper. In this thesis, beam search is run with the SubDisc default search width of $w = 100$ and a search depth of $d = 3$ with further configuration details described in Chapter 3.

2.5 Numeric Strategy

In SubDisc, numeric attributes are handled with a numeric strategy that determines how split points are evaluated for each candidate subgroup [8]. Four strategies are available: NUMERIC_BINS, which evaluates a fixed number of equally-spaced split points and retains all of them as candidates; NUMERIC_BEST_BINS, which evaluates the same equally spaced split points but retains only the single best split; NUMERIC_INTERVALS, which finds the optimal interval at depth 1 and reuses it at further depths, but is only valid for single nominal targets with specific quality measures; and NUMERIC_BEST, which exhaustively evaluates all possible split points and retains only the single best split for each candidate [8].

In this thesis, NUMERIC_BEST is used as it considers every possible split point, giving beam search the best possible numeric condition for each candidate without restricting to a fixed number of bins. However, since the best split is chosen independently for each candidate at each search level, the selected value reflects what is optimal given the current partial subgroup description, not the final set of conditions. This is the core limitation that the post-hoc optimisation methods in this thesis aim to address.

2.6 Search Methods

The four post-hoc search methods investigated in this thesis all operate on the subgroups returned by SubDisc, refining only the numeric condition values to improve WRAcc quality. Each method searches within a bounded interval around the original values produced by beam search, leaving categorical conditions unchanged. The methods differ in how thoroughly they explore this search space, each having a different trade-off between the quality of improvement achieved and the computational cost.

Search methods can broadly be divided into two categories [9]. Brute force methods evaluate candidates exhaustively and are guaranteed to find the best result within the search space, but become expensive as the search space grows. Heuristic methods instead use strategies to navigate the search space more efficiently, at the cost of not guaranteeing an optimal result. The four methods in this thesis cover both categories; full and interval brute force on the exhaustive side, and cross search and hill climbing on the heuristic side.

2.6.1 Brute Force

Full brute force exhaustively evaluates all possible combinations of numeric condition values across all unique values in the dataset [9]. This guarantees finding the optimal combination within the dataset, but at high computational cost. The complexity is k^n, where k is the number of unique values per condition and n is the number of numeric conditions, making it exponential in the number of conditions. For example, two numeric conditions each with 1000 unique values already result in one million combinations to evaluate. In this thesis, full brute force serves as an upper bound on achievable quality, allowing the heuristic methods to be evaluated against a guaranteed optimum.

2.6.2 Interval Brute force

Interval brute force applies the same exhaustive search as full brute force, but restricts the search space to a bounded interval around the original values produced by beam search [9]. The assumption is that beam search already produces values close to the optimum, making a full search of the entire value space unnecessary. By limiting the search to $p\%$ of unique values around the original value, the number of combinations is reduced significantly compared to full brute force. Since brute force complexity is exponential, even a small reduction in the search range per condition leads to a large reduction in total combinations evaluated. Interval brute force therefore offers a more computationally competitive alternative while still exhaustively covering the bounded search space.

2.6.3 Cross Search

Cross search is an axis-aligned local search method inspired by coordinate descent. In standard coordinate descent, each coordinate is optimized sequentially and repeatedly until convergence; meaning the result of optimizing one coordinate influences the next iteration [10]. Cross search differs in that it performs only a single pass: each numeric condition is optimized independently, keeping all other conditions fixed, and the best candidate found across all axes is returned as the final result. This means cross search does not iterate until convergence and cannot find

improvements that require changing multiple conditions at the same time. Compared to brute force, it evaluates far fewer combinations; for n conditions each with k possible values, brute force evaluates k^n combinations whereas cross search evaluates only nk. This comes with the cost of potentially missing joint optima.

2.6.4 Hill Climber

Hill climbing is, like cross search, an axis-aligned heuristic search method. The key difference lies in its iterative nature: rather than evaluating all candidates within an interval in a single pass, hill climbing moves one step at a time to an neighbouring candidate value for any condition, accepting the move only if it improves WRAcc. This process repeats until no neighbouring candidate has an improvement, at which point the method has converged to a local optimum. Compared to cross search and brute force, hill climbing typically evaluates the fewest combinations, as it terminates as soon as no local improvement is found. The trade-off for this is that it is the most susceptible to local optima, since it can never move through a non-improving step to reach a better solution elsewhere in the search space.

However, since the starting point for hill climbing is a subgroup already optimised by beam search, it is likely already close to a local optimum. This could make hill climbing an efficient choice in this context, as only a small number of steps may be needed to reach the true local optimum from the beam search result.

2.7 Related Work

Most existing work on improving subgroup quality focuses on the search process itself rather than refining results afterwards. Van Leeuwen and Knobbe introduce Diverse Subgroup Set Discovery (DSSD), which addresses redundancy by incorporating diversity strategies directly into beam search [3]. Rather than removing redundant subgroups post-hoc, DSSD prevents redundancy during search by selecting candidates that are both high quality and diverse in their subgroup covers, outperforming traditional SD methods particularly on more complex tasks.

More recently, García et al. propose DASSD, a heuristic algorithm that discovers high-quality, non-redundant subgroup sets by jointly optimising global set quality and redundancy during search [11]. Where DSSD reduces redundancy by promoting diversity among beam candidates, DASSD adapts the minimum quality threshold during search and uses a Minimum Redundancy Maximum Relevance (mRMR) pruning mechanism to select subgroups that are highly relevant and minimally redundant. Both approaches prevent redundancy during search, whereas this thesis applies redundancy filtering as a post-processing step after beam search has completed.

Besides work on redundancy, research has also been conducted on handling numeric conditions in SD. Meeng and Knobbe compare three search strategies for handling numeric attributes: traditional beam search, complete search, and cover-based subgroup selection [6]. Their experiments show that the choice of numeric strategy significantly affects subgroup quality, with more fine-grained threshold evaluation generally producing better results. This work differs from this thesis in that it focuses on the search process itself rather than applying post-hoc refinement to the results of an existing search.

An important consideration in subgroup discovery is whether the computational cost of a method is justified by the quality improvement it achieves. Van Leeuwen and Ukkonen argue that the quality-diversity trade-off is best addressed by explicitly considering the Pareto front of non-dominated solutions [12]. To the best of our knowledge, no existing work directly applies post-hoc local search methods to refine the numeric condition values of subgroups found by beam search. This thesis addresses this gap by investigating whether such refinement can improve subgroup quality, and how different methods compare in terms of the trade-off between quality gained and computational cost.

3 Experimental Setup

3.1 Datasets

The post-hoc search methods are evaluated on three publicly available datasets: Adult [13], German Credit [14], and Pima Indians Diabetes [15]. These datasets were selected to cover a range of dataset sizes, feature compositions, and application domains, allowing the effect of post-hoc optimisation to be analysed across varying experimental conditions. An overview of their key characteristics is shown in Table 1.

3.1.1 Adult

The Adult dataset was extracted by Barry Becker from the 1994 US Census database [13]. With 48,842 instances it is the largest dataset used in this thesis, and it contains six numeric and eight categorical features. The target variable is income, which is binary: leq50K or gr50K per year. The dataset is class-imbalanced, with 77% of instances belonging to the leq50K class. Three categorical features: workclass, occupation, and native-country, contain missing values, which are not fixed since the post-hoc methods in this thesis only modify numeric condition values.

3.1.2 German

The German Credit dataset was created by Hans Hofmann and contains 1,000 instances describing individuals applying for credit [14]. It contains seven numeric and thirteen categorical features. The target variable is credit risk, which is binary: good or bad. The dataset has a class imbalance, with 70% of instances classified as good credit risk and 30% as bad. The dataset contains no missing values.

3.1.3 Diabetes

The Pima Indians Diabetes dataset contains 768 instances describing female patients of Pima Native American heritage [15]. It contains eight numeric features and no categorical features. The target variable is diabetes diagnosis, which is binary: positive (1) or negative (0). The dataset has a class imbalance, with 65% of instances classified as negative and 35% as positive. While the dataset has no missing values, several features such as blood pressure and BMI contain zero values that are impossible, suggesting these represent missing data rather than true measurements. As

this thesis only optimises for WRAcc, the validity of the resulting conditions in the real world is not a concern.

3.1.4 Motivation

The three datasets were selected to provide as complete a review as possible across varying experimental conditions. Adult, German Credit, and Diabetes are well-established benchmark datasets with reliable quality. Together they offer variance in size, feature space, and domain. This allows the effect of post-hoc optimisation to be evaluated across diverse settings.

All three datasets have a binary target variable with class imbalance, where one class is dominant. This has been a deliberate choice. Subgroup Discovery aims to find subsets of the data that deviate from the overall population with respect to a target [7], making imbalanced datasets particularly interesting. This thesis considers finding which conditions cause a smaller group to deviate substantially from the dominant class, the kind of insight SD is designed to provide.

	Adult	**German Credit**	**Diabetes**
Instances	48,842	1,000	768
Features (excluding target)	14	20	8
Numeric features	6	7	8
Categorical features	8	13	0
Target	Income (leq50k/gr50k)	Credit risk (good/bad)	Diabetes (0/1)
Class balance	77% / 23%	70% / 30%	65% / 35%
Missing values	Yes	No	No (zeros as missing)
Domain	Census	Finance	Medical

Table 1: Overview of datasets used in the experiments.

3.2 Configuration

The experiments were implemented in Python using the pysubdisc wrapper for SubDisc [5]. All experiments were run on the LIACS computing cluster "drogium" which is equipped with 128 AMD EPYC 7702 cores at 2.00GHz (256 threads) and 512GB of RAM. Methods were parallelised across interval sizes and datasets using Python's `ProcessPoolExecutor`, with a maximum of 31 (10 intervals $\times$ 3 methods + 1 full Brute Force) workers running at the same time.

SubDisc was configured identically across all datasets in terms of search method, depth, width, and numeric strategy. The quality measure minimum was determined per dataset using SubDisc's significance threshold tool at a 5% significance level. The search depth and width are set to the SubDisc defaults of 3 and 100 respectively. The full configuration settings are shown in Table 2. The search methods were run with 10 intervals: from 5% until 50% with steps of 5% and one full brute force over the entire search space.

Setting	**Adult**	**German Credit**	**Diabetes**
Target value	gr50K	bad	1
Quality measure	WRAcc	WRAcc	WRAcc
Quality minimum	0.0263	0.0304	0.0287
Search depth	3	3	3
Search width	100	100	100
Numeric strategy	NUMERIC_BEST	NUMERIC_BEST	NUMERIC_BEST
Subgroups found	94	64	154

Table 2: SubDisc configuration per dataset.

3.3 Evaluation Metrics

The results are evaluated using three measures: WRAcc (see Section 2.2), runtime, and the number of redundant subgroups filtered. Mean WRAcc improvement is reported both before and after redundancy filtering, as a method may achieve high improvement on average, but based on subgroups that are later removed as redundant. Separating these two measures allows a fairer assessment of each method's performance. The WRAcc is also used for the 95th percentile and maximum improvement plots and the number of redundant subgroups filtered is shown in the Runtime vs Redundancy plot.

Table 3 provides an overview of the plots used to evaluate the results, listing each plot along with its axes. Runtime is plotted on a logarithmic scale, as the exponential growth of bruteforce runtime at larger intervals would otherwise diminish the differences between the faster methods: cross search and hill climbing, making them difficult to distinguish and analyse.

Plot	**X-axis**	**Y-axis**
Runtime vs Improvement (pre-filter)	Runtime (log)	Mean WRAcc improvement
Runtime vs Improvement (post-filter)	Runtime (log)	Mean WRAcc improvement
Runtime vs Redundancy	Runtime (log)	Subgroups filtered
Interval vs 95th Percentile	Interval size	95th percentile WRAcc improvement
Interval vs Maximum	Interval size	Maximum WRAcc improvement

Table 3: Overview of plots used to evaluate experimental results.

The runtime plots compare methods directly on the quality-cost trade-off, both before and after redundancy filtering to assess their improvements with and without filtering. The interval plots show how improvement scales with search space size, using the 95th percentile and maximum instead of the mean, to capture whether methods achieve large improvements on specific subgroup even when the average is modest.

A Pareto front is constructed over runtime and mean WRAcc improvement before and after redundancy filtering. Methods or configurations that fall below the Pareto front are dominated, meaning another method achieves equal or better improvement in less time, and can be considered objectively worse. The methods above the front are presented without further ranking, leaving the choice of preferred trade-off between quality and speed to the reader.

4 Results

4.1 Adult

4.1.1 SubDisc Baseline

SubDisc found 94 subgroups on the Adult dataset with a mean WRAcc of 0.076, ranging from 0.049 to 0.103. The majority of subgroups (73.4%) consist of three conditions and 97.9% contain at least one numeric condition, with a mean of 1.98 numeric conditions per subgroup. Subgroup coverage ranges from 195 to 616 instances, which is between 0.4% and 1.3% of the dataset, so the discovered subgroups describe small but distinct segments of the population.

4.1.2 Comparing Results for Methods

Figure 1 shows the post-hoc optimisation results for the Adult dataset before redundancy filtering. Mean WRAcc improvement ranges from 0.74% for hill climbing at the 5% interval to 2.83% for full brute force. Hill climbing improves from 0.74% at the 5% interval to 0.89% at the 10% interval, after which it plateaus completely. Cross search shows a clear increase from 0.87% at the 5% interval to 2.31% at the 35% interval, after which it plateaus. Interval brute force follows the same pattern, increasing from 1.00% at the 5% interval to 2.82% at the 35% interval, plateauing and matching full brute force from that point onward.

Hill climbing completes its search in under a second across all intervals, while cross search ranges from 3 to 19 seconds. Interval brute force and full brute force are considerably more expensive, ranging from 31 seconds at the 5% interval to over 4.7 hours for full brute force.

Regarding the Pareto front, for hill climbing all intervals above 10% are dominated. For cross search, the 5% interval and all intervals above 35% are dominated. For interval brute force, intervals below 20% and above 35% are dominated. These intervals do not offer a better trade-off between improvement and runtime than the non-dominated alternatives.

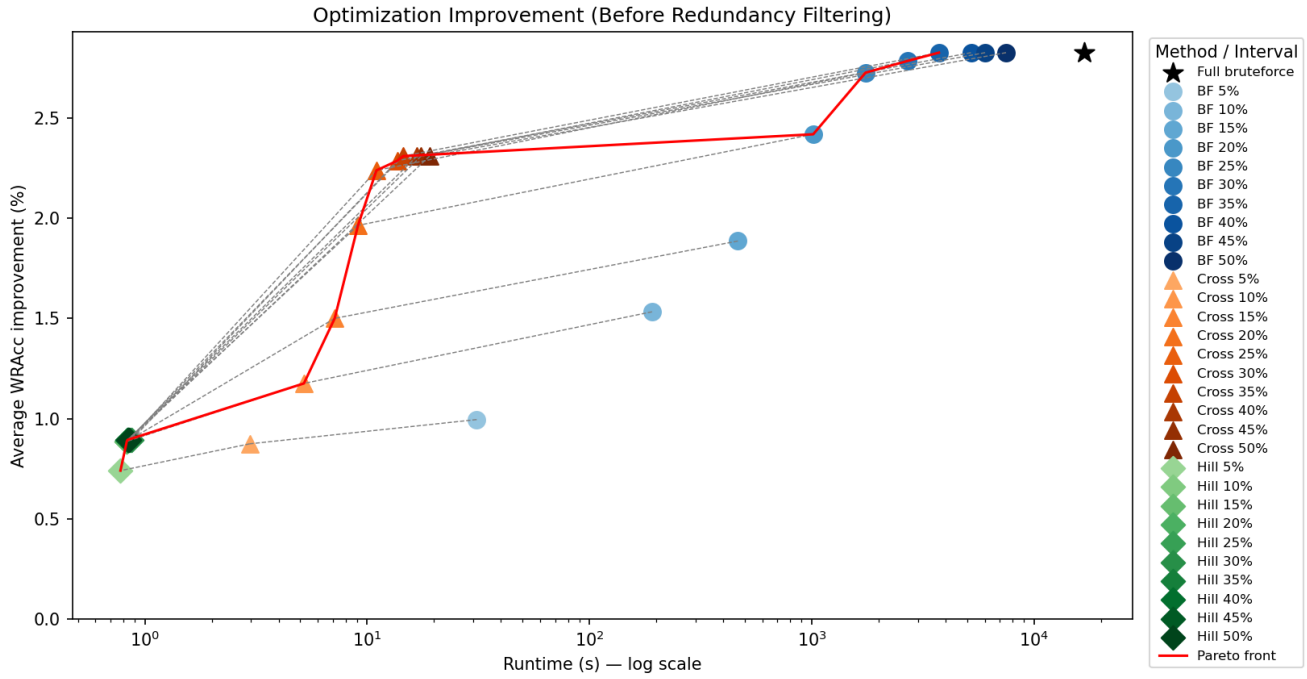

Figure 1: Optimization improvement before redundancy filtering for Adult dataset.

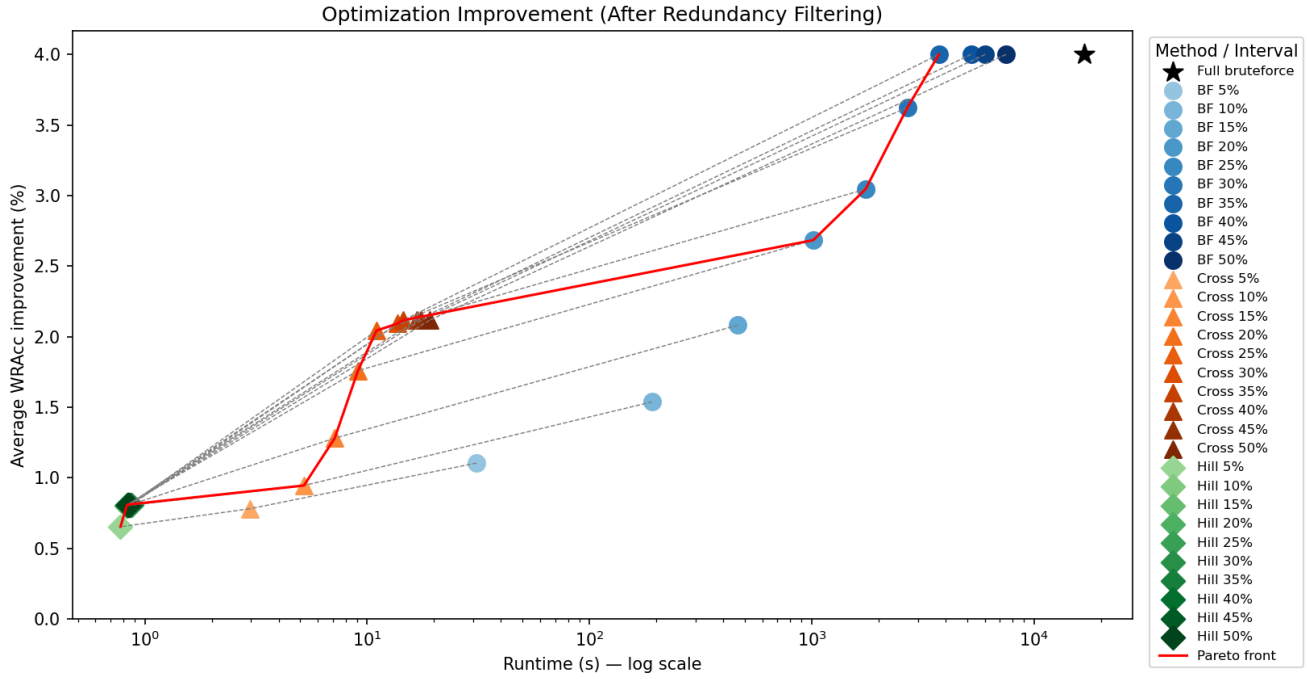

Figure 2: Optimization improvement after redundancy filtering for Adult dataset.

The effect of redundancy filtering is shown in Figure 2. Hill climbing improves 16 subgroups per interval with 2 removed, showing a small decrease in mean improvement after filtering. Cross search improves between 26 and 31 subgroups depending on the interval, with 2 subgroups removed at 5% and 3 removed for all intervals above, also showing a small decrease. Interval brute force and full brute force show a notable increase in mean improvement after filtering. The highest mean improvement rises from 2.83% before filtering to 4.0% after filtering for brute force from 35% onward. The differences between the 20% and 35% intervals also become larger after filtering, growing from 0.40% before filtering to 1.32% after. As shown in Figure 3, brute force removes substantially more redundant subgroups than the heuristic methods, improving between 16 and 27 subgroups while removing between 7 and 21.

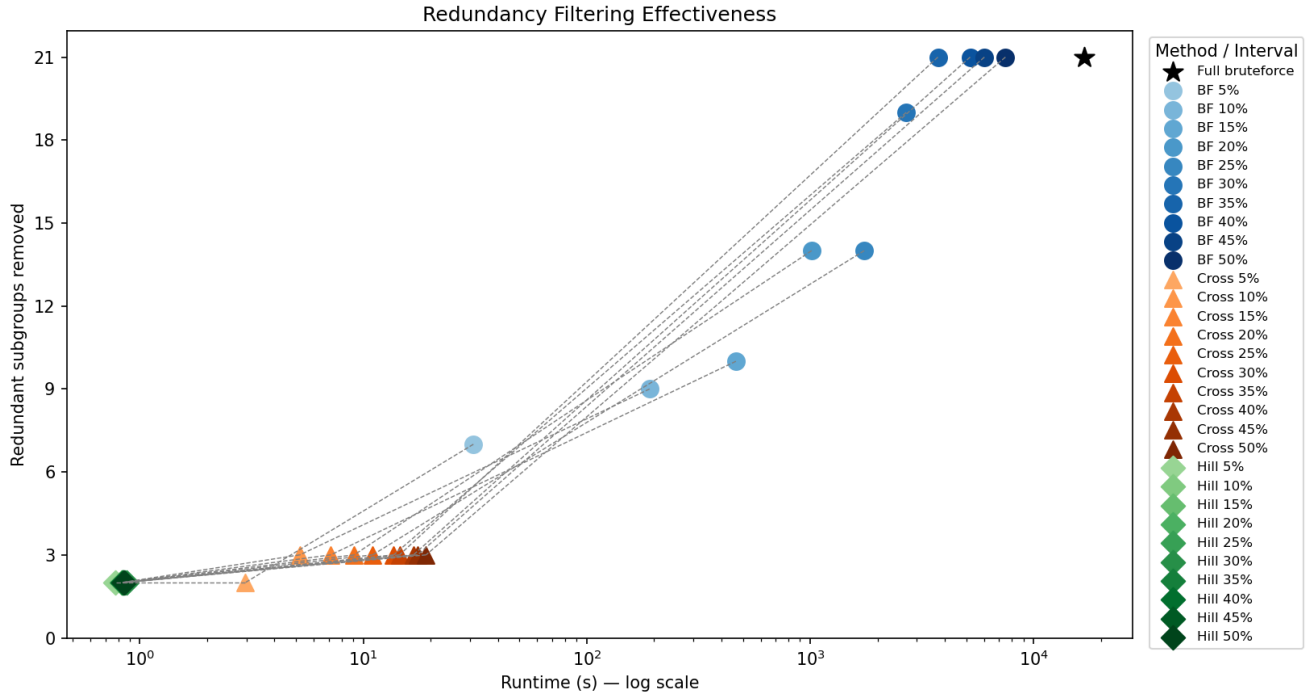

Figure 3: Redundancy filtering for Adult dataset.

Figure 4 shows the 95th percentile and maximum improvement per method across interval sizes after filtering. Hill climbing shows the same behaviour as in the mean improvement plots, improving from 5% to 10% and plateauing immediately after, with the maximum improvement remaining flat across all intervals.

Cross search and interval brute force show more varied behaviour compared to the mean improvement plots. In the 95th percentile plot, interval brute force briefly exceeds the full brute force reference line at the 25% interval, but drops back below it at larger intervals. From 25% onward, interval brute force reaches 24.69% at the 95th percentile and 26.30% for the best subgroup, while cross search reaches 23.93% and 25.22% respectively, both substantially higher than the mean improvement of 2.83%. In both plots, interval brute force matches full brute force from the 30% interval, while cross search approaches but never fully reaches it.

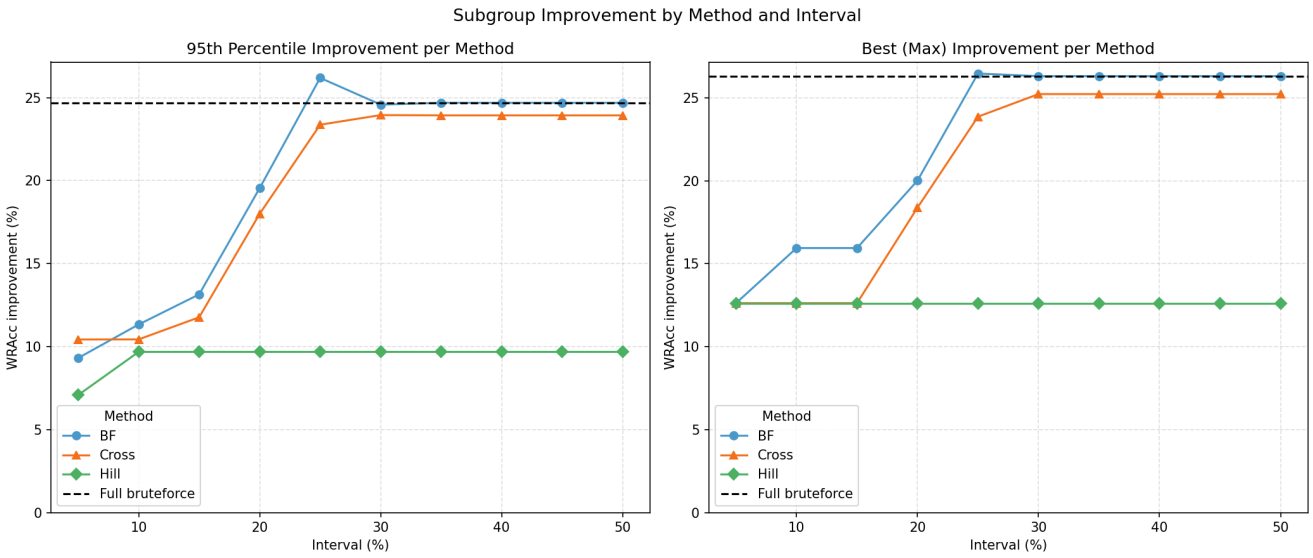

Figure 4: 95th Percentile and Maximum Improvement for Adult dataset.

4.2 German

4.2.1 SubDisc Baseline

SubDisc found 64 subgroups on the German Credit dataset with a mean WRAcc of 0.046, ranging from 0.031 to 0.058. The majority of subgroups (64.1%) consist of three conditions, and 93.8% contain at least one numeric condition, with a mean of 1.52 numeric conditions per subgroup. Subgroup coverage ranges from 226 to 603 instances, representing between 22.6% and 60.3% of the dataset.

4.2.2 Comparing Results for Methods

Figure 5 shows the post-hoc optimisation results for the German Credit dataset before redundancy filtering. Overall improvement is minimal across all methods, with mean WRAcc improvement ranging from 0.024% for hill climbing to 0.53% for full brute force. Hill climbing plateaus immediately at the 5% interval and shows no further improvement across any interval. Cross search similarly plateaus at the 5% interval at 0.054% improvement, with runtime increasing but no additional improvement gained. Interval brute force matches cross search up to 20%, after which it shows gradual improvement through larger intervals, reaching 0.43% at 50%. It does not converge with full brute force which achieved 0.54% mean improvement.

Hill climbing completes its search in under half a second across all intervals, while cross search ranges from 2 to 13 seconds. Brute force ranges from 11 seconds at the 5% interval to just under an hour for full brute force.

Regarding the Pareto front, the non-dominated configurations are hill climbing at 5%, cross search at 5%, interval brute force from 25% onward with the exception of 30%, and full brute force. All other configurations are dominated.

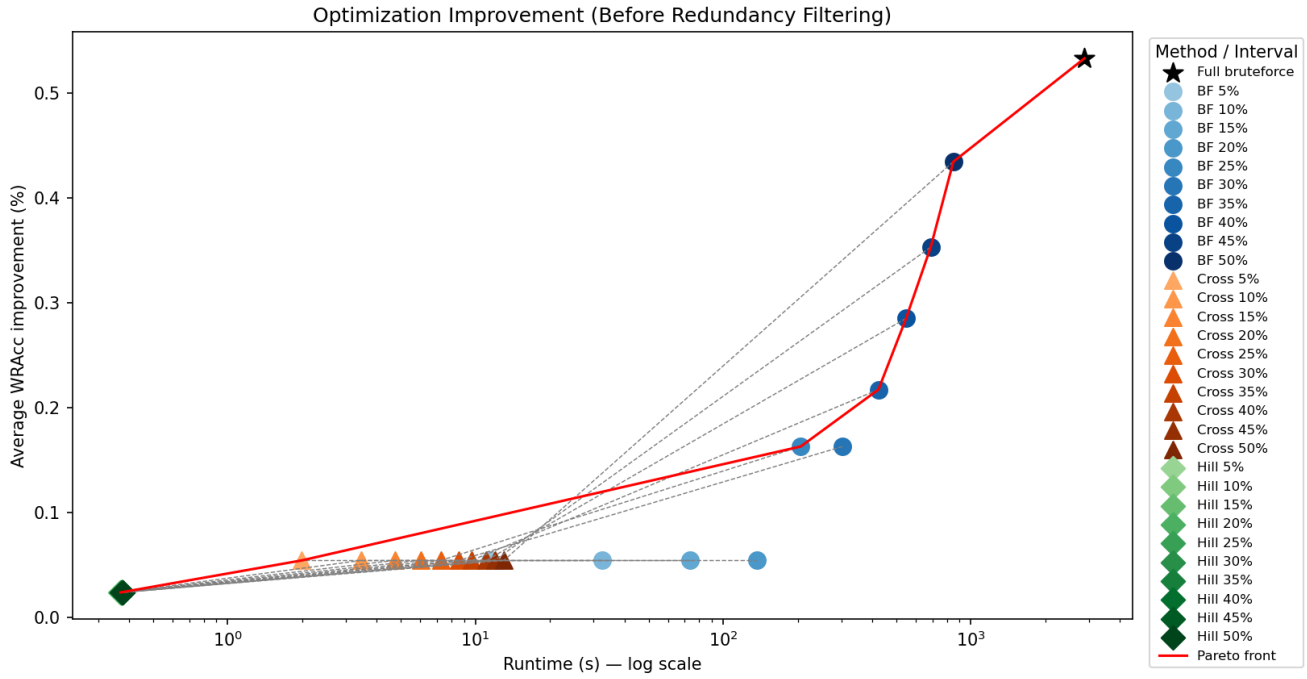

Figure 5: Optimization improvement before redundancy filtering for German dataset.

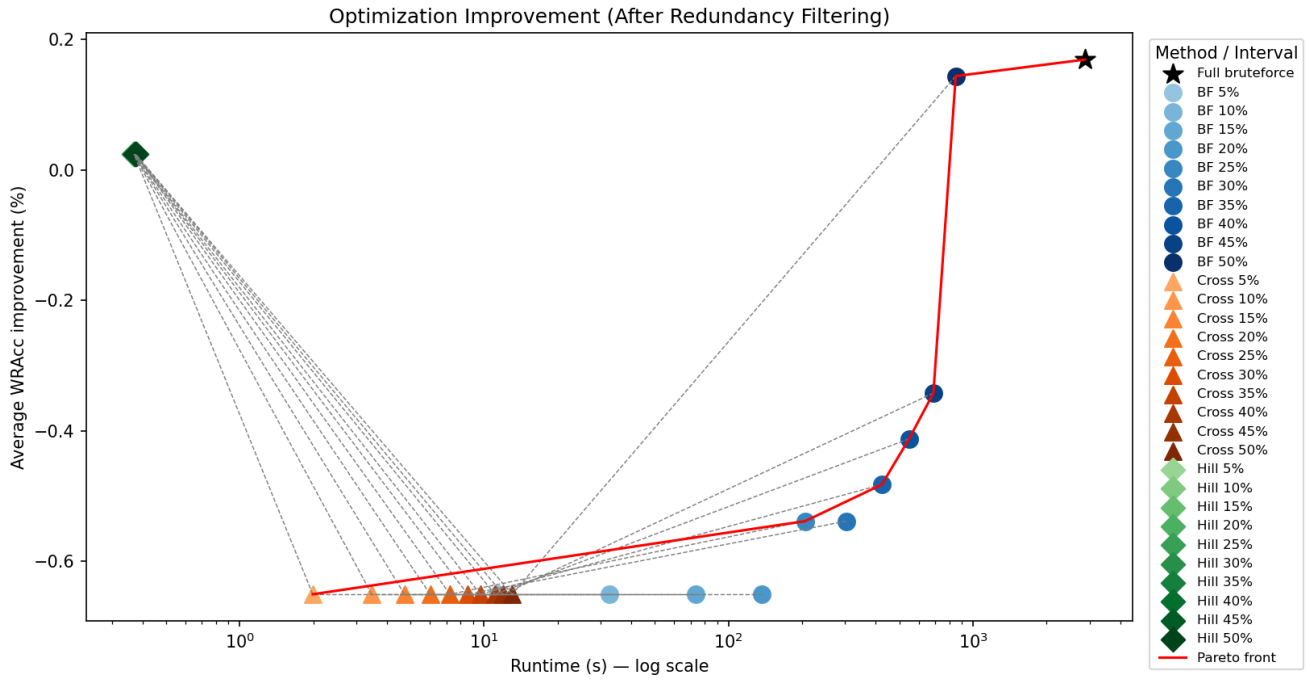

Figure 6: Optimization improvement after redundancy filtering for German dataset.

Figure 6 shows a notably different picture after redundancy filtering. Every method and interval combination shows a decrease in mean WRAcc improvement except hill climbing, which is unaffected since it produces no redundant subgroups and improves only 2 subgroups per interval, see Figure 7. Every cross search interval has 2 subgroups removed and 3 improved. Brute force intervals up to 45% also have 2 subgroups removed with 3 to 4 improved, while BF 50% and full brute force have 4 removed and 2 to 3 improved out of 64 total. These removals resulted in a mean WRAcc below the original baseline from SubDisc for all cross search and most brute force intervals.

Hill climbing is excluded from the Pareto front in Figure 6 since it removes no redundant subgroups, leaving its mean WRAcc artificially higher than methods that do filter. This is discussed in section 5.2. The dominated and non-dominated methods remain the same as before filtering when hill climbing is excluded from the comparison.

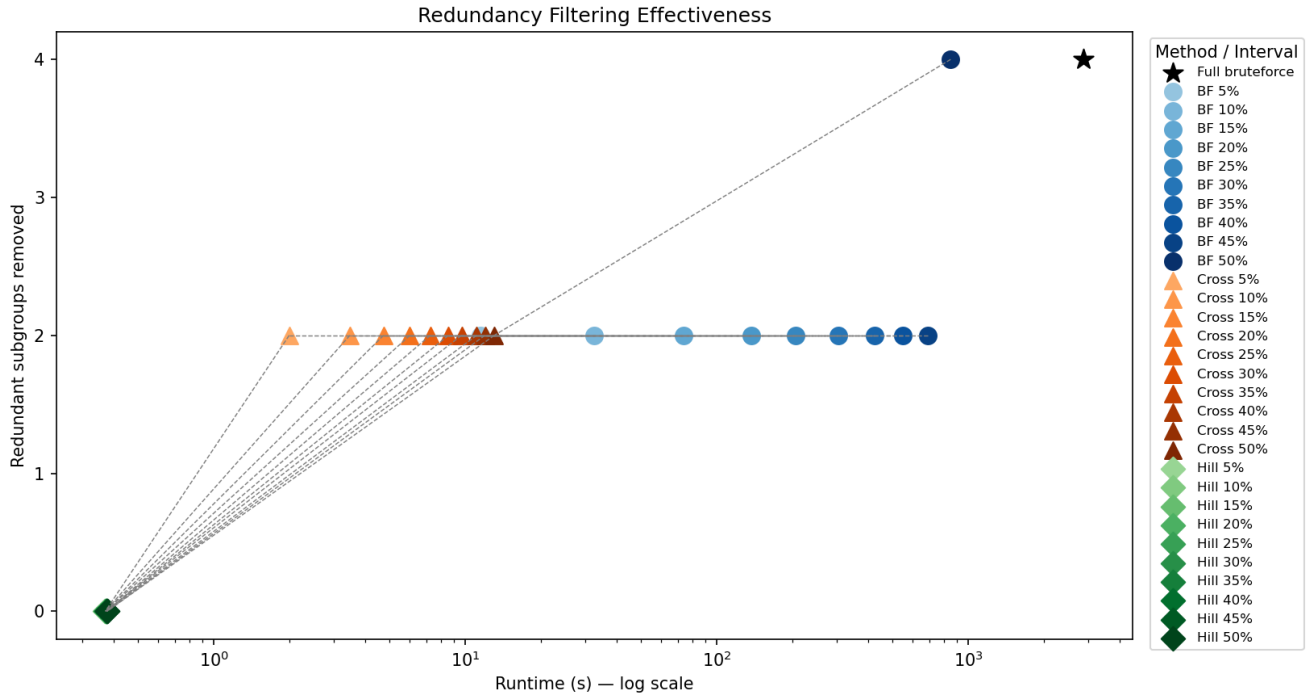

Figure 7: Redundancy filtering for German dataset.

Figure 8 shows that the 95th percentile and maximum improvement plots follow the same pattern of ups and downs, with the maximum values slightly higher than the 95th percentile. The hill climber and cross search both plateau at 1.7% immediately, never increasing their improvement. Interval brute force finds increasing improvement from 20% to 45%, after which there is a steep decline that goes below the improvement of the hill climber and cross search. This decline at 50% is likely caused by the redundancy filter removing the subgroup responsible for the high individual improvements at larger intervals, though this was not directly verified.

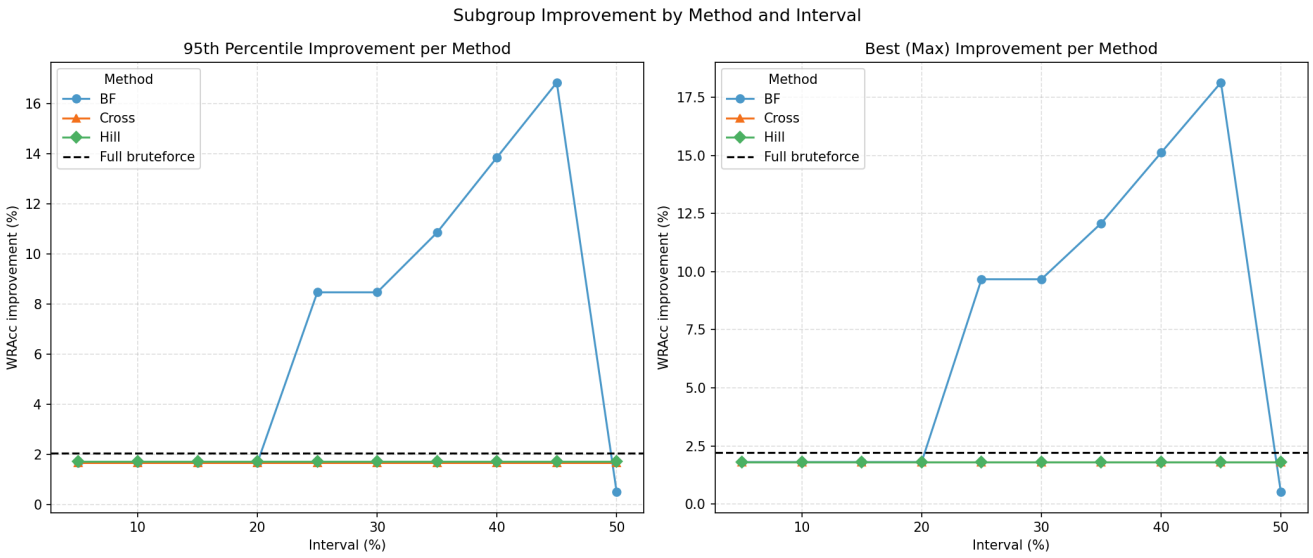

Figure 8: 95th Percentile and Maximum Improvement for German dataset.

4.3 Diabetes

4.3.1 SubDisc Baseline

SubDisc found 154 subgroups on the Diabetes dataset with a mean WRAcc of 0.069, ranging from 0.048 to 0.106. The majority of subgroups (79.2%) consist of three conditions, and all subgroups contain exclusively numeric conditions, reflecting the fully numeric feature space. Subgroup coverage ranges from 114 to 477 instances, representing between 14.8% and 62.1% of the dataset.

4.3.2 Comparing Results for Methods

The results before redundancy filtering for the Diabetes dataset are shown in Figure 9. Mean WRAcc improvement ranges from 0.88% for hill climbing at the 5% interval to 23.33% for full brute force. Hill climbing improves from 0.88% at the 5% interval to 2.83% at the 25% interval, after which it plateaus completely. Cross search shows an increase from 1.30% at the 5% interval to 13.89% at the 50% interval, never fully plateauing. Interval brute force increases from 1.33% at the 5% interval to 21.40% at the 50% interval, and full brute force achieves a mean improvement of 23.33%.

Hill climbing completes the search in under 2 seconds across all intervals, while cross search ranges from 7 to 45 seconds. Brute force is considerably more expensive, ranging from 7 minutes at the 5% interval to over 53 hours for the 50% interval.

The resulting Pareto front has some notable features. For hill climbing, only the 10%, 15% and 40% intervals are non-dominated. The 40% interval achieves equal or higher improvement than all intervals from 20% onward in a shorter runtime. For cross search, only the 5% interval is dominated by hill climbing, the rest is non-dominated. Only 2 interval brute force intervals are non-dominated: 45% and 50%, with full brute force also being non-dominated.

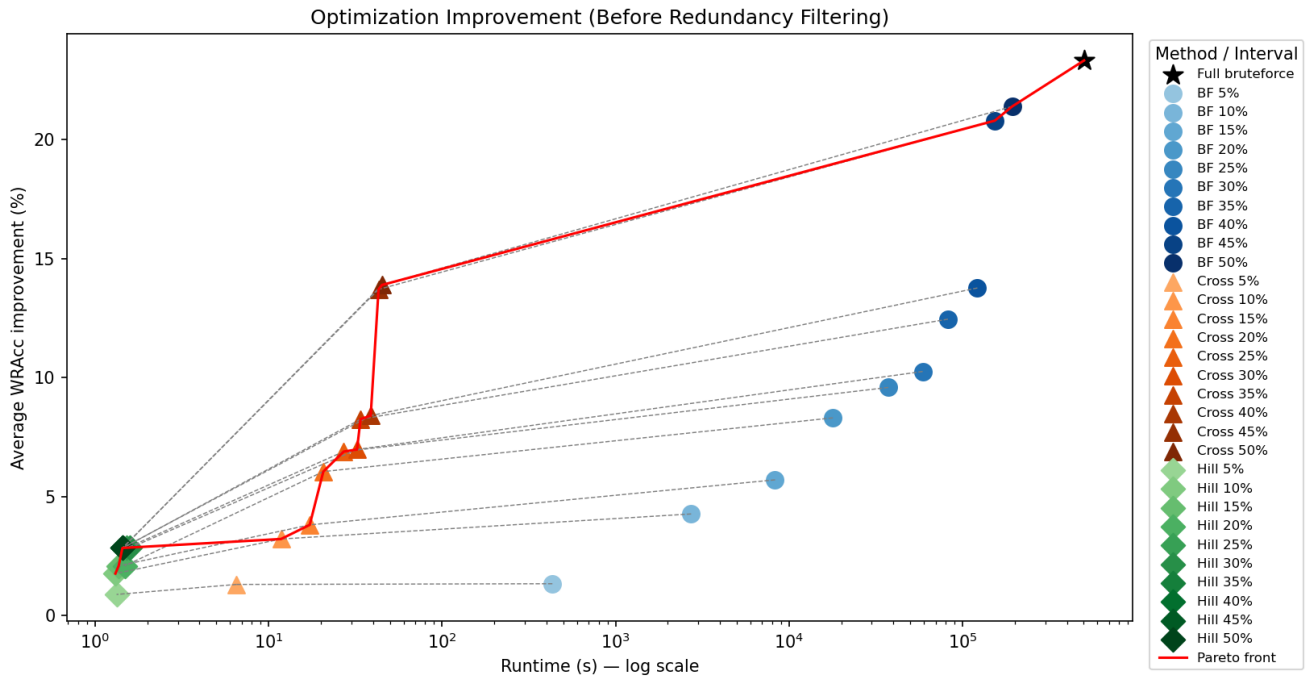

Figure 9: Optimization improvement before redundancy filtering for Diabetes dataset.

The results after applying redundancy filtering are shown in Figure 10. For hill climbing and cross search the impact is minimal, with mean improvement dropping by at most 0.18% and 0.09% respectively, as only 2 and 1 subgroups are removed per interval. For brute force the impact is clearly visible, with the mean improvement dropping by as much as 6.49% at the 50% interval. The Pareto front is almost identical to before filtering, with the only change being that BF 50% is now dominated by BF 45%.

These changes are consistent with the number of subgroups removed shown in Figure 11. Every hill climbing interval has 2 subgroups removed and 46 improved. Every cross search interval has 1 subgroup removed, with between 79 and 108 subgroups improved depending on the interval. For brute force, the number of subgroups removed grows from 4 at the 5% interval to 63 at 50%, with between 48 and 91 subgroups improved, and full brute force removing 68 and improving 43 out of 154 total.

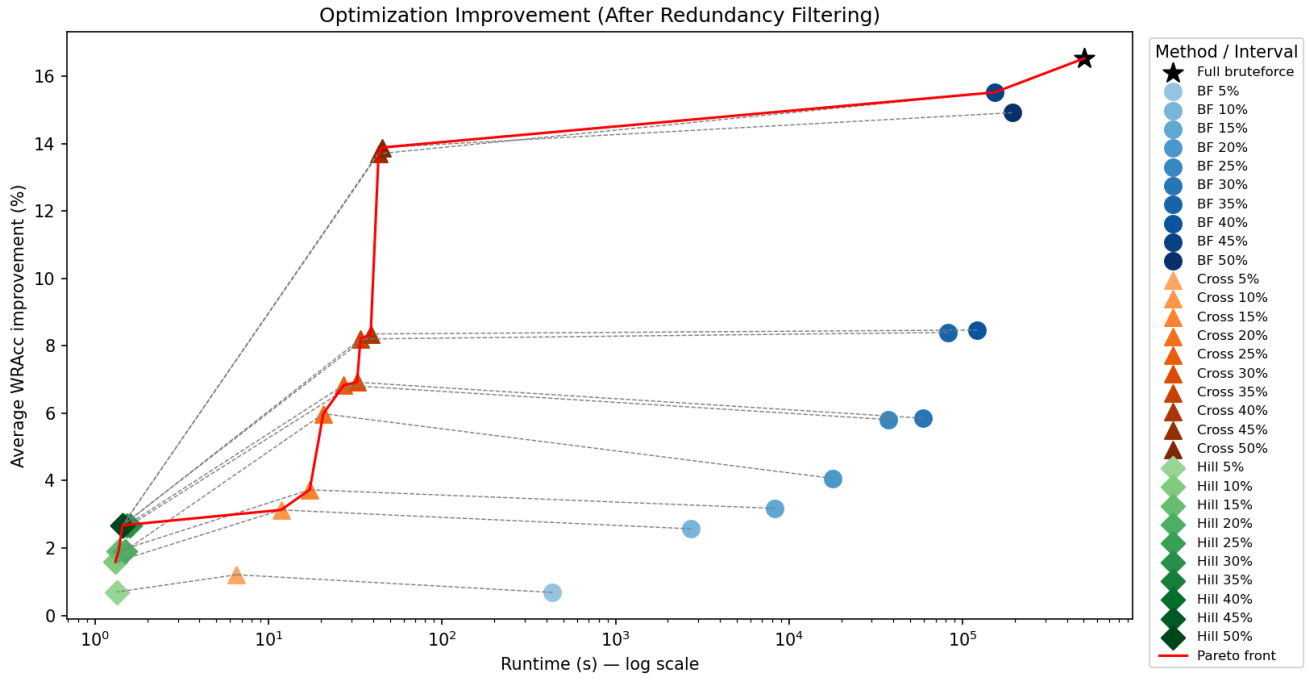

Figure 10: Optimization improvement after redundancy filtering for Diabetes dataset.

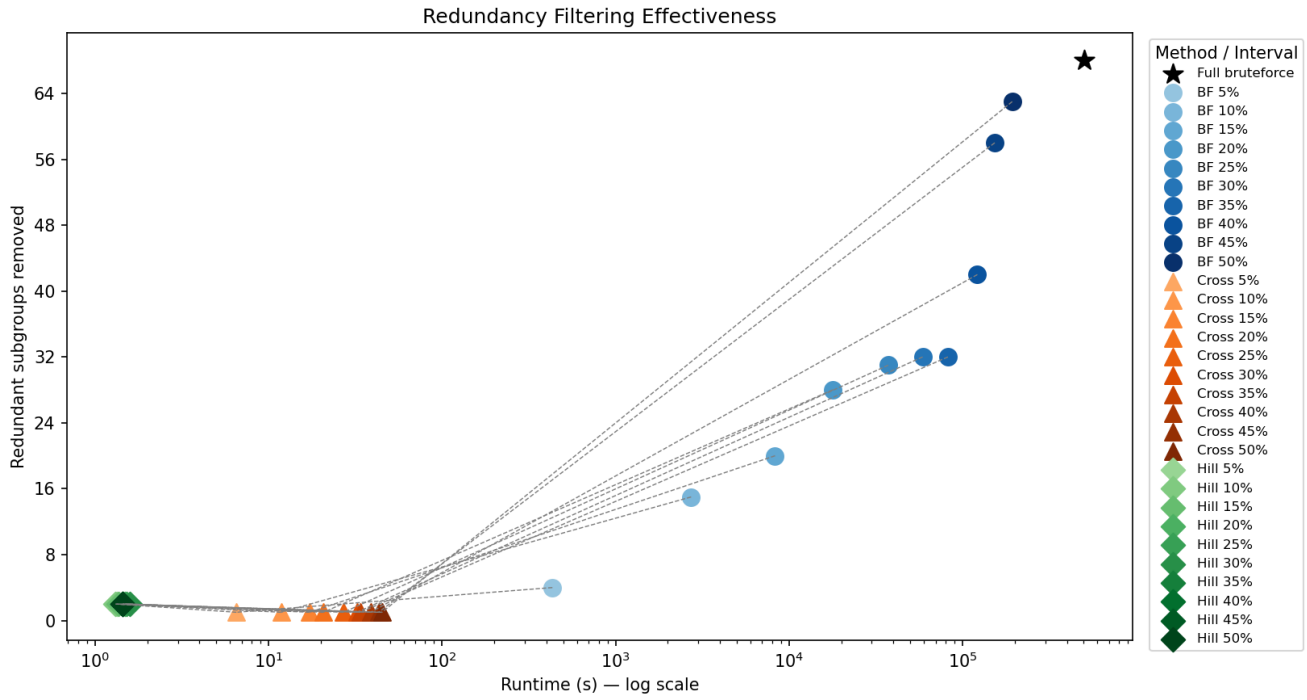

Figure 11: Redundancy filtering for Diabetes dataset.

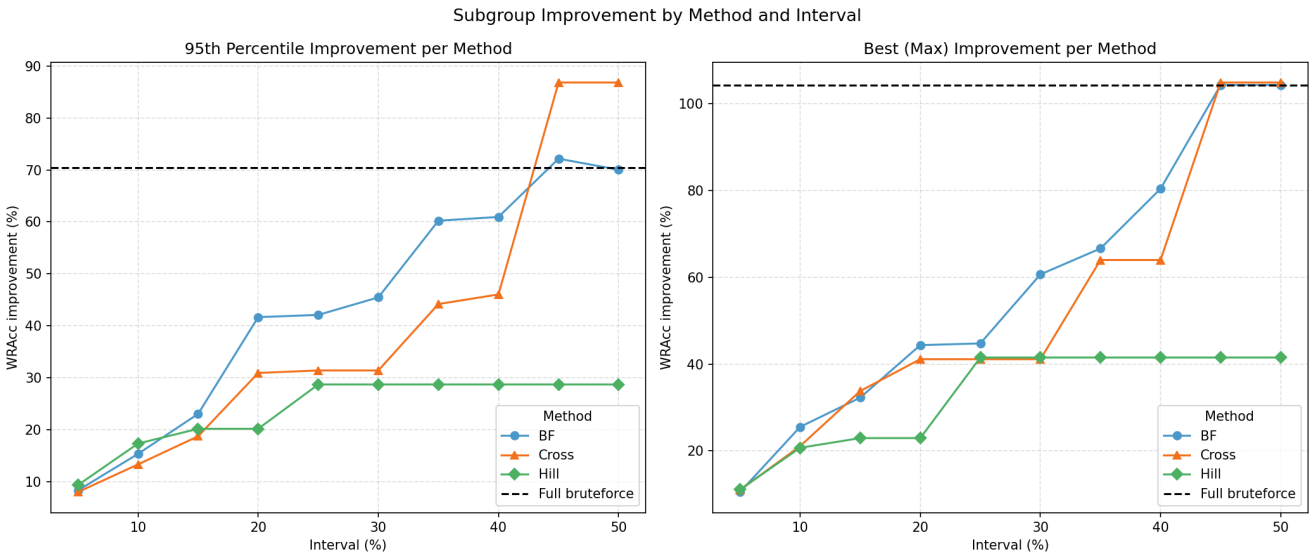

Figure 12: 95th Percentile and Maximum Improvement for Diabetes dataset.

Figure 12 shows the 95th percentile and maximum improvement per method across interval sizes after filtering. Hill climbing plateaus at the 25% interval in both plots, consistent with the mean improvement results. The maximum improvement it achieves of around 41% is substantially higher than its mean improvement of 2.83%.

Cross search and interval brute force show more varied behaviour compared to the mean improvement plots. Cross search shows a sharp jump at the 45% interval to 87% at the 95th percentile and 105% for the best subgroup. Interval brute force grows more steadily across all intervals, reaching around 70% at the 95th percentile and matches cross search at 105% for the best individual subgroup at 45% and 50%.

5 Discussion and Conclusion

The results show varied outcomes across datasets, with mean WRAcc improvements and the number of subgroups removed differing per dataset. Notable differences were also observed between the methods and the interval sizes used. In this section, we address each sub-question in turn, discussing the results per method, offering possible explanations, and concluding with a summary. After all sub-questions have been addressed, we summarize the findings to provide a final comparison and answer the main research question.

5.1 Improvement across Intervals

For this sub-question, mean WRAcc improvement is reported before redundancy filtering to ensure the filtering process does not influence the comparison between methods and interval sizes.

5.1.1 Hill Climber

Hill climbing achieved the lowest mean WRAcc improvement of the three methods across all datasets. For Adult and German Credit, the highest mean improvements were 0.89% and 0.024% respectively, while Diabetes showed a higher ceiling of 2.83%. In all three datasets, hill climbing reaches its plateau early: at the 10% interval for Adult, the 5% interval for German Credit, and the 25% interval for Diabetes, gaining nothing but more runtime from that point. Hill climbing was outperformed by cross search and interval brute force on all intervals in German Credit and nearly all intervals on Adult and Diabetes, with only the 5% cross search interval falling below it.

The plateauing after relatively small intervals is consistent with hill climbing's local search nature; since it only moves to neighbouring candidate values and stops as soon as no local improvement is found, it converges quickly and leaves most of the additional interval unexplored. The beam search starting point is likely already close to a local optimum, so hill climbing only needs a small search space around it to find it. Although hill climbing does not appear to require a large interval, providing one carries no additional computational cost, as the extra search space simply goes unexplored. Therefore, it is recommended to always provide a large interval to ensure the optimal local solution is reached.

5.1.2 Cross Search

Cross search achieved moderate mean WRAcc improvements across all datasets, almost always outperforming hill climbing while always remaining below interval brute force when given the same interval. For Adult, mean improvement increased from 0.87% to 2.31% at the 35% interval, after which it plateaued. For German Credit, cross search plateaued immediately at 0.054%. Diabetes showed the strongest results, with improvement increasing from 1.30% to 13.89% at the 50% interval, never fully plateauing.

The plateau behaviour on Adult and German Credit, and the continued improvement on Diabetes, may be related to cross search's single-pass nature. It optimises each numeric condition independently and returns the best candidate found across all axes. However, since this evaluates each condition in isolation, it cannot find joint optima. Once the best value configuration achievable in a single pass has been found, expanding the interval further yields no additional gain, as any further improvement would require changing multiple conditions simultaneously. The continued improvement on Diabetes may be explained by the larger numeric feature space meaning the single-pass optimum has not yet been reached within the search range.

5.1.3 Brute Force

Interval brute force achieved the highest mean WRAcc improvements of the three interval-based methods across all datasets. For Adult, mean improvement increased from 1.00% to 2.82% at the 35% interval, plateauing and matching full brute force from that point onward. For German Credit, improvement grew gradually from the 20% interval, reaching 0.43% at the 50% interval but never converging with full brute force which achieved 0.54%. Diabetes showed the strongest results, with improvement increasing from 1.33% to 21.40% at the 50% interval, never plateauing. Full brute force serves as the upper bound on achievable quality, reaching 2.83% on Adult, 0.54% on German

Credit, and 23.33% on Diabetes.

Since interval brute force exhaustively evaluates all combinations of numeric condition values within the bounded search space, it is guaranteed to find the best possible configuration within that range, allowing it to find joint optima that cross search and hill climbing cannot.

The convergence of interval brute force with full brute force at 35% on Adult suggests that beam search with NUMERIC_BEST already produces values close to the true optimum on this dataset, and the optimal configuration lies within a relatively small range of the starting values. For German Credit and Diabetes, no convergence was observed even at the 50% interval, suggesting the optimal configuration lies further from the beam search starting point on these datasets. The reasons for this difference were not directly investigated in this thesis.

5.1.4 Possible Influence of the Datasets

The differences in mean WRAcc improvements for all the methods and their intervals across the datasets may be related to data characteristics. Diabetes contains exclusively numeric features, and with more numeric conditions per subgroup, there are more candidates to evaluate, giving the methods more room to find improvements. Adult and German Credit contain fewer numeric conditions per subgroup on average, which may limit the improvement they can achieve.

The minimal improvement on German Credit may be related to its large subgroup coverage of 22 to 60% of the dataset. When a subgroup covers a large portion of the data, a small numeric threshold adjustment changes relatively few instances, producing a negligible change in the target rate and therefore in WRAcc. This is consistent with interval brute force only starting to outperform the heuristic methods from the 30% interval onward on German Credit. It could be that it is only possible at these larger intervals to make bigger threshold changes across multiple conditions that produce a meaningful change in coverage. However, the Diabetes dataset has comparable subgroup coverage and shows notable improvement across all methods, so subgroup size alone is not a sufficient explanation, and these remain possible explanations rather than confirmed causes.

5.1.5 Conclusion

In conclusion, for hill climbing a large interval is unnecessary since it plateaus early, but providing one carries no additional cost as the extra space goes unexplored. Cross search benefits consistently from larger intervals, making larger intervals the better choice when using this method. For interval brute force, a larger interval is preferable as smaller intervals can achieve similar improvement to cross search, making the exhaustive search only worthwhile when a sufficiently large interval is used. Full brute force always achieves the highest improvement as it exhaustively searches the entire value space, making it the best choice when maximising quality is the sole concern.

5.2 Redundancy Filtering

In the results, redundancy filtering caused a decrease in mean WRAcc improvement for some methods and datasets. This should not be interpreted as filtering hurting performance. Rather,

the post-filter mean WRAcc is a more accurate reflection of the actual unique subgroup quality, as the pre-filter mean can be inflated by redundant subgroups that represent the same underlying pattern at different specificity levels. A drop in mean WRAcc after filtering therefore indicates that some of the apparent improvement was due to redundancy rather than genuine discovery of diverse high-quality subgroups.

5.2.1 Hill Climber

Hill climbing was least affected by redundancy filtering, with only two, zero, and two subgroups removed for Adult, German Credit, and Diabetes respectively, consistent across all intervals. This is likely due to its local search nature: since hill climbing converges to the nearest local optimum from its starting point, two subgroups can only converge to the same configuration if their starting values already lie within the same local basin. As most SubDisc subgroups have different starting values, convergence is rare and filtering rates remain low regardless of interval size.

5.2.2 Cross Search

Cross search shows a similar filtering rate to hill climbing, with three, two, and one subgroups removed for Adult, German Credit, and Diabetes respectively. Despite exploring a wider range of values than hill climbing, the number of subgroups filtered remains low across all datasets and intervals. This is likely due to its single-pass nature; since cross search fixes all other conditions while optimizing each one independently, the optimal value found for any condition depends on the current values of all other conditions. Two subgroups with different starting values will therefore likely find different optimal values for each condition, making full convergence across all conditions unlikely. The interval only had an effect on adult where 5% filtered two and the rest three. For the other datasets it did not have any effect on the filtering.

5.2.3 Brute Force

Interval brute force showed substantially higher filtering rates than the heuristic methods, with the number of subgroups removed growing with interval size. At the largest intervals, up to 21, 4, and 63 subgroups were removed for Adult, German Credit, and Diabetes respectively. For Adult and German Credit, the filtering rate reaches its maximum at around 35% and 25% respectively, after which it remains stable. For Diabetes, the filtering rate continues to increase until full brute force which removes 68 subgroups out of 154.

5.2.4 Datasets

The filtering rates across datasets are largely driven by the amount of improvement each dataset allows. For German Credit, only a small number of subgroups are improved across all methods, leaving very few candidates that could converge to the same configuration, resulting in low filtering rates. For Adult, the moderate improvement leads to moderate filtering, particularly for brute force where the larger search space increases the chance of convergence. The presence of categorical conditions in Adult additionally limits convergence, as subgroups differing in categorical values can never converge regardless of how their numeric values are optimised. For Diabetes, the substantially higher improvement across all methods leads to the highest filtering rates, particularly for interval

brute force. The exclusively numeric feature space means all subgroup conditions are subject to optimisation, also increasing the likelihood that different subgroups converge to the same optimal configuration when exhaustively searched.

5.2.5 Conclusion

The heuristic search methods show substantially lower filtering rates than interval brute force across all datasets and intervals. If preserving a diverse set of subgroups is important, the heuristic methods are preferable as they rarely cause convergence and therefore retain more of the original subgroup set. Interval brute force on the other hand produces the most filtering, which can be desirable when the goal is a clean non-redundant set of only the highest quality subgroups.

5.3 Cost vs Quality Trade-Off

For this sub-question, once again mean WRAcc improvement is reported before redundancy filtering to ensure the filtering process does not influence the comparison between methods and interval sizes.

5.3.1 Hill Climbing

Hill climbing is by far the fastest method, completing in under 1 second on Adult and German Credit and under 2 seconds on Diabetes, while still achieving modest improvement. It has one to three non-dominated intervals depending on the dataset, reflecting its favourable runtime-to-improvement ratio. Although its absolute improvement is lower than cross search and interval brute force, the negligible computational cost makes it an attractive first choice when time is limited.

An interesting observation occurs in the Diabetes dataset, where the 40% interval appears on the Pareto front ahead of the 20% to 35% intervals. Since hill climbing converges quickly and evaluates very few candidates regardless of interval size, the runtimes are so small that server variance likely influences the measured values. In theory, the 25% interval would be the non-dominated configuration at that runtime range rather than the 40% interval.

5.3.2 Cross Search

With the exception of the 5% interval on Adult and Diabetes, cross search is non-dominated on the Pareto front. On German Credit, only the 5% interval is non-dominated, as improvement does not increase beyond that point while runtime continues to grow, meaning all larger intervals are dominated by the 5% interval itself. On Diabetes, cross search is non-dominated across nearly all intervals, reflecting its continued improvement with larger search spaces as discussed in Section 5.1. Where cross search is non-dominated, it offers a strong trade-off between runtime and improvement, completing in under 50 seconds across all intervals and datasets, making it a practical choice when higher improvement than hill climbing is needed but full brute force is computationally infeasible.

5.3.3 Brute Force

Interval brute force achieves the highest mean WRAcc improvement among the interval-based methods, but this comes at a significant computational cost. At smaller intervals, it is dominated on the Pareto front, offering no better improvement than cross search at a substantially higher runtime. At larger intervals it becomes non-dominated, but the additional improvement gained over cross search is modest relative to the large increase in runtime. Runtime ranges from 31 seconds to 4.7 hours on Adult, from 11 seconds to just under an hour on German Credit, and from 7 minutes to nearly 6 days on Diabetes. Full brute force consistently achieves the highest improvement of all methods but is only justified when quality is the sole concern and computational cost is no constraint, as its runtime is orders of magnitude higher than the interval-based methods.

5.3.4 Conclusion

The three methods offer different trade-offs between computational cost and quality improvement. Hill climbing is the fastest option and is recommended when a quick, low-cost improvement is sufficient, as it converges in seconds while still achieving modest gains. Cross search offers a balanced trade-off, achieving substantially higher improvement than hill climbing while keeping runtime within a manageable range. The optimal interval depends on the dataset; on Adult, 25 to 35% offers the best balance, while on Diabetes larger intervals continue to gain improvement.

Interval brute force offers the highest improvement of the interval-based methods but at exponentially higher computational cost. It is only recommended when maximising quality is the primary concern and computational cost is not a limiting factor. Full brute force should only be considered when an absolute upper bound on quality is required, regardless of runtime.

5.4 Summary and Main Research Question

In this thesis, the effects of three post-hoc optimisation methods: hill climbing, cross search, and interval brute force, were investigated across three datasets and ten interval sizes. All four post-hoc methods were found to be capable of improving subgroup quality to varying degrees, with the amount of improvement depending on the method, interval size, and dataset characteristics. The key findings per method are summarised below.

Hill climbing is the fastest method and is well-suited for situations where a quick, low-cost improvement of the beam search results is sufficient. Its performance and runtime are consistent across datasets. Since hill climbing rarely causes subgroups with the same conditions to converge to the same configuration, it preserves the diversity of the subgroup set. This makes it preferable when retaining multiple configurations of the same condition set is desirable, but less suitable when only the highest quality configuration per condition set is needed.

Cross search offers the best balance between improvement and computational cost. It achieves substantially higher improvement than hill climbing while completing in under a minute across all datasets and intervals, and similarly to hill climbing it rarely causes convergence, preserving diversity in the subgroup set. Since cross search only evaluates nk candidates rather than the full search space, and is anchored to the beam search starting point, it is considered a genuine

post-hoc refinement rather than a rediscovery of results that a broader search would have found. Its single-pass nature limits it to axis-aligned improvements, meaning it cannot find joint optima, but this also keeps its runtime manageable. Cross search is recommended as the default post-hoc method when both quality improvement and runtime are important considerations.

Interval brute force achieves the highest improvement of the interval-based methods but at exponentially higher computational cost. It is not recommended for most applications, as cross search achieves comparable improvement at a fraction of the runtime. However, it is the preferred choice when even marginal quality gains matter, or when a clean non-redundant set of only the highest quality configurations is desired, as it produces the most redundancy filtering. At larger intervals, interval brute force approaches a full search of the value space, raising the question of whether it still constitutes genuine post-hoc optimisation or effectively rediscovers results that a broader beam search would have found. Full brute force serves as the theoretical upper bound on achievable quality but is only justifiable when runtime is no constraint.

The results suggest that datasets with a higher proportion of numeric features and more numeric conditions per subgroup offer more room for post-hoc optimisation, as seen with Diabetes outperforming Adult and German Credit across all methods. For datasets with a smaller proportion of numeric conditions, such as German Credit, the computational cost of post-hoc optimisation is unlikely to be justified by the quality gains achieved. However, since only one dataset of each type was evaluated, these findings cannot be generalised and serve as preliminary observations only.

6 Further Research

This thesis evaluated the post-hoc methods on three datasets with different characteristics, which limits the generalisability of the findings. The strong contrasts between the datasets raise the question of whether the improvement observed on each is representative of a broader class of datasets or an outlier. Testing on more datasets with similar characteristics would allow investigation of whether the post-hoc optimisation performance generalises.

A natural extension of cross search is full coordinate descent, which iterates over all conditions repeatedly until convergence rather than making a single pass. This would allow improvements that require changing multiple conditions to be found, potentially closing the gap with interval brute force at a lower computational cost than exhaustive search.

This thesis only optimised numeric condition values, leaving categorical conditions unchanged. Extending the post-hoc methods to also refine categorical conditions could yield additional improvements, though changing a categorical value fundamentally alters the meaning of a condition rather than refining it, raising the question of whether this still constitutes post-hoc optimisation or a rediscovery of different subgroups.

WRAcc was used as the quality measure throughout this thesis. Investigating whether the findings hold for other quality measures would broaden the applicability of the results.
Finally, this thesis used the SubDisc default search width of $w = 100$ and depth of $d = 3$. Investi-

gating the effect of different search widths and depths on the starting quality of the subgroups, and consequently on the room available for post-hoc improvement, would provide additional insight into the interaction between beam search configuration and post-hoc optimisation.

References

[1] H.M. Proença, P. Grünwald, T. Bäck, and M. van Leeuwen. Robust subgroup discovery: Discovering subgroup lists using mdl. *Data Mining and Knowledge Discovery*, 36(5):1885–1970, August 2022.

[2] Martin Atzmueller. Subgroup discovery. *WIREs Data Mining and Knowledge Discovery*, 5(1):35–49, 2015.

[3] M. van Leeuwen and A.J. Knobbe. Non-redundant subgroup discovery in large and complex data. In Dimitrios Gunopulos, Thomas Hofmann, Donato Malerba, and Michalis Vazirgiannis, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 459–474, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[4] Nada Lavrač, Branko Kavšek, Peter Flach, and Ljupčo Todorovski. Subgroup discovery with cn2-sd. *J. Mach. Learn. Res.*, 5:153–188, December 2004.

[5] A.J. Knobbe, J. Gobeil, M.K. van Dijk, and W.J. Palenstijn. Subdisc. https://github.com/SubDisc/SubDisc, 2021.

[6] Marvin Meeng and Arno Knobbe. For real: a thorough look at numeric attributes in subgroup discovery. *Data Mining and Knowledge Discovery*, 35:158–212, 2021.

[7] Marvin Meeng, Wouter Duivesteijn, and Arno Knobbe. *ROC search — An ROC-guided Search Strategy for Subgroup Discovery*, pages 704–712. 04 2014.

[8] SubDisc. Numeric strategy. https://github.com/SubDisc/SubDisc/wiki/Numeric-Strategy, 2024.

[9] Judea Pearl and Richard E. Korf. Search techniques. *Annual Review of Computer Science*, 2:451–467, 1987.

[10] Stephen J. Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34, 2015.

[11] Aarón García, Guillermo Vigueras, and Alfonso Mateos. Dassd: Dynamic and adaptive subgroup set discovery with redundancy control. *Information Sciences*, 749:123531, 2026.

[12] Matthijs van Leeuwen and Antti Ukkonen. Discovering skylines of subgroup sets. In *Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2013*, volume 8190 of *Lecture Notes in Computer Science*, pages 272–287, Berlin, Heidelberg, 2013. Springer.

[13] Barry Becker and Ronny Kohavi. Adult. https://archive.ics.uci.edu/dataset/2/adult, 1996.

[14] Hans Hofmann. Statlog (german credit data). https://archive.ics.uci.edu/dataset/144/statlog+german+credit+data, 1994.

[15] J.W. Smith, J.E. Everhart, W.C. Dickson, W.C. Knowler, and R.S. Johannes. Pima indians diabetes database. https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database, 1988.