Communication-less Strategies for Widening

Dissertation zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften (Dr. rer. nat.)

vorgelegt von

Violeta N. Ivanova-Rohling

an der

Universität Konstanz
Mathematisch-Naturwissenschaftliche Sektion
Fachbereich für Informatik und Informationswissenschaft

Tag der mündlichen Prüfung: 3. Mai 2019
1. Referent: Prof. Dr. Michael Berthold
2. Referent: Prof. Dr. Sven Kosub

Contents

Zusammenfassung

Der Trend, die Anzahl der Kernen anstelle der Schaltzyklen zu erhöhen, hat in den vergangenen Jahren die Bedeutung der Entwicklung paralleler Algorithmen in allen Bereichen der Informatik gesteigert. Dieser Trend ist auch an dem Gebiet des Data-Mining nicht vorbei gegangen. Das Wachstum der Datengrößen kombiniert mit den immer weiter wachsenden parallelen Rechenressourcen hat den Fokus der Forschung im parallelen Data-Mining auf das Design effizienterer Algorithmen gelenkt. Diese Dissertation hat einen anderen Blickwinkel auf paralleles Data-Mining: anstatt die parallele Version eines Algorithmus zu nutzen, um eine Lösung der selben Qualität zu erhalten, nur auf schneller Art und Weise, soll untersucht werden, wie parallele Rechenressourcen die Qualität der Lösung verbessern können. Oft wenden Data-Mining-Algorithmen eine Greedy-Heuristik an, welche auf lokaler Optimalität basiert, um die Suche durch einen riesigen Raum potentieller Lösungen zu ermöglichen. Aufgrund dieser eingeschränkten Erforschung des Lösungsraumes ist das Auffinden der optimalen Lösung nicht garantiert. Der Fokus dieser Arbeit ist die Entwicklung von Strategien zum Investieren paralleler Rechenressourcen für die Verbesserung der Ergebnisse, die die standardmäßige Greedy-Data-Mining-Heuristik liefert, indem die parallelen Ressourcen zur besseren Erforschung des Suchraums (der Modelle) eingesetzt werden. Dieser Ansatz wird als *Widening* einer Suchheuristik bezeichnet. Zusätzlich soll die Laufzeit des Algorithmus so nah wie möglich an der Laufzeit des ursprünglichen Algorithmus bleiben. Die meisten auf der Greedy-Suche basierenden Data-Mining-Algorithmen können als eine Iteration eines Verfeinerungs- und eines Auswahloperators, welche auf intermediäre Modelle angewendet werden bis ein Modell gefunden wurde, welches gewisse Kriterien erfüllt, dargestellt werden. Ein einzelner Schritt dieser Heuristik hat die Form:

$$m' = s\left(r(m)\right)$$

Die beiden Funktionen $s(\cdot)$ und $r(\cdot)$ beschreiben einen *Auswahl-* beziehungsweise einen *Verfeinerungsoperator*. Widening ist dann definiert als

$$M' = \{m'_1, \ldots, m'_k\} = s_{\mathrm{W}}\left(\bigcup_{m \in M} r(m)\right).$$

In jedem Schritt betrachtet der Auswahloperator s_{W} die Verfeinerung einer Menge M mit Kardinalität k ursprünglicher Modelle und gibt eine neue Menge M' von k verfeinerten Modellen für weitere Untersuchungen aus. Hier wird k als die *Breite* des Widening bezeichnet. Die trivialste Implementierung von Widening ist die einfache Beam-Search, $Top-k$ Widening. Dieser Ansatz verlangt jedoch die ständige Synchronisation zwischen den parallelen Prozessen. Um den Mehraufwand dieser Kommunikation zu vermeiden, richtet sich der Fokus dieser Dissertation auf Widening-Strategien, welche den Suchraum auf strukturierte Weise ohne die Notwendigkeit von Kommunikation erforschen.

Ideales Widening wird definiert als theoretischer Rahmen zum Widening. Dabei erfolgt eine explizite Partitionierung des Suchraums mit angestrebten Eigenschaften. Jede Partition wird von einem parallelen Prozess erforscht. Ideales Widening ist jedoch in der Praxis für eine allgemeinen Art von Suchräumen schwer zu erreichen. Daher werden hier verschiedene praxistaugliche Ansätze für kommunikationsfreies Widening betrachtet. Diese Ansätze basieren auf impliziter Partitionierung des Suchraums durch die für jeden parallelen Prozess individualisierte Modifizierung des Auswahloperators. Ein anderes zu berücksichtigendes Problem der Widening-Ansätzen ist die Gefahr der Konvergenz zu einem lokalem Optimum. Dies ist ein bekanntes Problem für die Ansätze, die auf Beam-Search basieren, und wird durch das Erzwingen von Diversität gelöst. Diversität bei der Erforschung des Suchraums ohne Kommunikation und ohne vorherige Kenntnis dieses Suchraums zu erreichen, ist nicht trivial. Ein einfacher Ansatz ordnet jedem parallelem Prozess individuelle Präferenzen bei der Modellauswahl zu. Dieser Ansatz ist parameterabhängig und erlaubt keine strukturierte Erforschung des Suchraums, was aber das Ziel von Widening ist. In dieser Dissertation wird ein auf Umgebungen basierender Ansatz definiert, *Widening durch Umgebungen*, welcher das Ziel der strukturierten Erforschung des Suchraums verfolgt. Widening durch Umgebungen definiert Umgebungen des lokal optimalen Modells mit verschiedenen Eigenschaften und partitioniert sie unter den parallelen Prozessen, wobei Labels verwendet werden, die vor Beginn der Suche zugeordnet werden. Drei Arten von Umgebungen werden untersucht: Optimalitätsumgebungen, für die die Metrik auf dem Qualitätsmaß der Modelle basiert; Ähnlichkeitsumgebungen, für die die Metrik eine Art von Modell- oder Daten-basiertem Abstandsmaß ist; und Diversitätsumgebungen, welche k Modelle, die sowohl vielfältig als auch von hoher Qualität sind, enthält. Verschiedene Ansätze zu den Diversitätsumgebungen werden in dieser Arbeit diskutiert–basierend auf einfachen Mindestabständen und solche, die durch das Suchen nach Nischen in evolutionären Algorithmen inspiriert sind. Unter festen Annahmen über den Suchraum lassen sich theoretische Eigenschaften der auf Umgebungen basierten Widening-Ansätze zeigen. Widening durch Optimalitätsumgebungen kann $Top-k$ Widening ohne die Notwendigkeit von Kommunikation emulieren, sofern die Anzahl der parallelen Prozesse, k, groß ist. Widening durch Ähnlichkeitsumgebungen kann genutzt werden um vielversprechende Gebiete des Suchraums zu untersuchen und für eine hinreichend große Anzahl paralleler Ressourcen, k, kann garantiert werden, dass die gefundene Lösung höchstens einen Abstand δ von der besten Lösung im untersuchten

Intervall entfernt ist. Widening durch Diversitätsumgebungen hilft der möglichen Konvergenz zu lokalen Optima zu entgehen, indem die parallelen Prozesse gezwungen werden Lösungen zu erforschen, die sowohl vielfältig als auch vielversprechend sind. Es erfolgt eine Untersuchung des Suchraums für die einfachste Art des Verfeinerungsoperators inklusive des Beweises, dass es sich bei diesem Raum um einen Verband handelt. Aus dieser Verbandseigenschaft, insbesondere der Isomorphie zum Verband der Teiler, folgt eine Möglichkeit den Suchraum unter den parallelen Prozessen zu partitionieren. In der vorliegenden Arbeit finden diese Widening-Ansätze Anwendung auf zwei Algorithmen, den Greedy-Algorithmus für das Mengenüberdeckungsproblem und den CN2-Algorithmus für Regelinduktion. Die zentralen Ergebnisse bestätigen die theoretischen Vorhersagen.

Alle Widening-Ansätze zeigen eine Verbesserung in der Qualität der Lösung im Vergleich zur Greedy-Lösung. Eine Erhöhung der Anzahl der parallelen Prozesse verbessert die Qualität der Lösung. Auch die angemessene Verwendung von Diversität verbessert das Ergebnis. Darüber hinaus, kann das Hinzufügen weiterer paralleler Prozesse den Mangel an Kommunikation kompensieren. Größere Umgebungen führen nicht zu verbesserten Ergebnissen. Diversität in Kombination mit Anforderungen an die Qualität der Modelle, wodurch die Untersuchung vielversprechender Lösungen erreicht wird, ermöglicht jedoch eine Verbesserung der Qualität der gefundenen Lösung am Ende des Algorithmus. Was das Widening durch Ähnlichkeitsumgebungen betrifft, ist es meist vorteilhaft kleine Umgebungen zu verwenden, dann das Ziel ist die Erschließung, das heißt die Suche nach ähnlichen Lösungen. Selbst für Widening durch Diversitätsumgebungen ist eine große Umgebung nicht immer gegenüber einer kleineren im Vorteil. Der Grund ist die Tatsache, dass diese Methode den Suchraum sehr dünn besetzt. Eine große Anzahl paralleler Prozesse und Widening mit einer kleineren Umgebung kann genügen um die wichtigen Peaks im Suchraum zu entdecken. Diversität hat, falls sie auf geeignete Weise bestimmt wird, das Potential die Qualität der Lösung zu verbessern.

Widening durch Optimalitätsumgebungen hat Laufzeiten, die denen des Greedy-Algorithmus sehr nahe kommen. Widening durch Diversitätsumgebungen ist am rechenintensivsten, wobei der wichtigste Faktor die Größe der Umgebung ist. Sowohl Widening durch Ähnlichkeitsumgebungen als auch Widening durch Diversitätsumgebungen benötigen eine Vorverarbeitung um ähnliche Laufzeiten wie der Greedy-Algorithmus zu erzielen. Voraussetzung ist das Vorhandensein von genügend parallele Ressourcen.

Summary

We live in the age of ever-increasing parallel computing resources, and as a consequence, for a decade there has been intense research into parallel data mining algorithms. Most of this research is focused on improving the running time of existing algorithms. In this dissertation we want to look at parallel data mining from a different perspective: instead of using the parallel versions of algorithms to obtain the solution with the same quality, only faster, we want to know how to invest parallel compute resources in a way which improves the quality of the solution. Because the space of potential solutions if typically enormous, and it cannot be explored through exhaustive search to find the optimal solution, often the data mining algorithms use a heuristic (for example a greedy search) in order to find a sufficiently good solution in a reasonable time. Due to this limited exploration, finding the optimal solution is not guaranteed. The focus of this work is to develop strategies for investing parallel compute resources to improve the result obtained by standard greedy data mining heuristics by investing parallel resources into better exploration of the (model) search space. We call this approach *Widening* of search heuristics. Additionally, we want to keep the running time of the algorithm as close to the running time of the original algorithm, as possible. Most greedy search based data mining algorithms can be represented as an iteration of a refinement and selection operator, which are applied to intermediate models until either a model is found that fits some criteria. A single step of the heuristics is expressed as follows:

$$m' = s\left(r(m)\right)$$

The two functions $s(\cdot)$ and $r(\cdot)$ describe a *selection* and a *refinement* operator, respectively. Widening is then defined as

$$M' = \{m'_1, \ldots, m'_k\} = s_{\mathrm{W}}\left(\bigcup_{m \in M} r(m)\right).$$

At each step, the selection operator s_{W} considers the refinements of a set M with cardinality k of original models and returns a new set M' of k refined models for further investigation. We will refer to k as the *width* of the Widening. The most trivial implementation of Widening is the simple beam search, $Top - k$ Widening. This approach,

however, requires continuous synchronization between the parallel workers. In order to avoid the consequent communication overhead, we are focused on Widening strategies, which explore the search space in a structured fashion, without the need for communication.

We describe *Ideal Widening*, a theoretical framework for Widening, defined as an explicit partition of the search space with desired properties. Each partition is assigned for exploration to a given parallel worker. However, Ideal Widening is difficult to achieve in practice for a general type of the search space. This is why we look at different communication-less Widening approaches, achievable in practice. These are typically based on the implicit partitioning of the search space via individualized modification of the selection operator for each parallel worker. Another problem of the Widening approaches to consider is the danger of converging to a local optimum. This is a well known problem for the beam-search-based approaches and is solved by enforcing diversity. Achieving diversity of exploration without communication and without prior knowledge of the search space is not trivial. We introduce a simple approach, which assigns individual model preferences to each parallel worker, prior to the beginning of the search. This approach is parameter dependent and does not allow for structured exploration of the search space, which is the goal of Widening. We define a neighborhood-based approach, *Widening via neighborhoods*, which aims at a structured exploration of the search space. Widening via neighborhoods defines neighborhoods of the locally optimal model with different properties and partitions them among the parallel workers, using labels assigned prior to the search. Three types of neighborhoods are investigated: optimality neighborhoods, for which the metric is based on the model quality measure; similarity neighborhoods, for which the metric is some type of model-based or data-based distance measure, and diversity neighborhoods, which contain diverse-and-good k models. Different approaches to diverse neighborhoods are described – some are based on simple diversity thresholds, others use more sophisticated strategies such as *nicheing.* We demonstrate theoretical properties of the neighborhood-based Widening approach, under fixed assumptions about the search space. Widening via optimality neighborhoods can emulate $Top - k$ Widening, without the necessity of communication, for a large enough number k of parallel workers. Widening via similarity neighborhoods can be used for exploitation of promising areas of the search space, and for a large enough number of parallel resources k, it can guarantee that the solutions discovered will be at most distance δ from the best solution in the investigated interval. Widening via diverse neighborhoods helps overcome a potential convergence to local optima, by forcing the parallel workers to explore diverse and promising solutions. We investigate the search space structure for the simplest type of algorithms and prove that it is a lattice. We use the properties of the search space to partition it among parallel workers. We demonstrate the benefits of and compare all these Widening strategies using two algorithms, the greedy algorithm for the set cover problem and the CN2 algorithm for rule induction. The main experimental results confirm the theoretical results.

All Widening approaches show improvement of the solution quality, when compared with the greedy solution. Increasing the number of parallel workers improves the solution quality. The appropriate use of diversity improves the solution quality. Additionally, adding more parallel workers can compensate for the lack of communication. A larger size of the neighborhood does not lead to improved results. However, diversity in combination with model quality, which leads to the investigation of promising solutions, provides an improvement of model quality. When it comes to Widening via similarity neighborhoods, it is most often more advantageous to use a small size of the neighborhood, since the goal is exploitation or similarity search. Even for Widening via diverse neighborhoods, a larger neighborhood size is not always more advantageous than a smaller one, due to the fact that this method is very sparse. Large number of parallel workers and Widening with a smaller neighborhood size can be sufficient for discovering the important peaks. Diversity has the potential to improve the quality of the solution, if it is appropriately selected. Widening via optimality neighborhoods has a running time close to that of the greedy algorithm. Widening via diversity neighborhoods is the most computationally intensive, with the biggest factor being the size of the neighborhood. Both, Widening via similarity neighborhoods and Widening via diversity neighborhoods need preprocessing in order to have a running time similar to that of the greedy algorithm, given sufficient parallel resources.

Chapter 1

Introduction

This chapter is adapted from [86], [16].

1.1 Background.

In the past decades the advances in parallel computer architecture brought about tremendous abundance of compute resources. These trends shifted the focus in most computational fields towards the development of parallel algorithms. Consecutively, developing parallel alternatives of sequential algorithms has become a dominant theme in data mining research as well. Parallel and more efficient versions of sequential algorithms already exist for most types of data mining algorithms. Additionally, more and more data from every aspect of human activity is being acquired, generated and stored ever more cheaply. This newly available possibility to generate, collect and store enormous amounts of data has led to an intense research focus on processing and analyzing these growing data repositories using distributed methods. These factors have brought about the *Big Data* paradigm. Following the new reality, most research in parallel data mining focuses either on the processing of larger data sets or on the speed-up of existing algorithms and much-needed advances have been made in this field. However, irrespective of how large the collected data, or how efficient the algorithm, the most critical goal of data mining is the high quality of the obtained solution. Namely, appropriate smart algorithms are needed to learn from the collected data in a correct way. Many problems exist for which a speedily obtained solution is not what is needed, but which instead require a solution of as high quality as possible. Since, in essence, there is no restriction on the availability of parallel resources, they can be used to discover better solutions to complex problems, than is possible with the existing heuristics. This is why, in contrast to most current directions in the field, our research focuses on investing parallel resources to obtain a solution of higher quality. Due to the enormous search space, the typical data mining algorithms cede search completeness by employing a heuristic to look for a solution in a feasible time. Often, data mining algorithms employ a greedy heuristic, which relies on the local optimality property to find a good solution. Due to this limited exploration, for most problems, finding the optimal solution is not guaranteed.

In this dissertation, we present strategies for investing parallel compute resources to improve the result obtained by standard greedy data mining heuristics. Our goal is to improve solution quality without increasing the overall time spent, by investing parallel resources into a better exploration of the (model) search space. Our aim is to reduce the influence of the heuristic and produce the same type of model but at a substantially higher level of quality. We achieve this by *widening* the search. Instead of pursuing only one solution at each step, we consider several solution candidates. The most trivial implementation of Widening is the simple beam search. This approach, however, requires continuous synchronization and brings about communication overhead, which in turn will penalize the running time of a given heuristic, and will violate our ambition of keeping it constant. That is why we investigate approaches that do not require communication between the parallel workers.

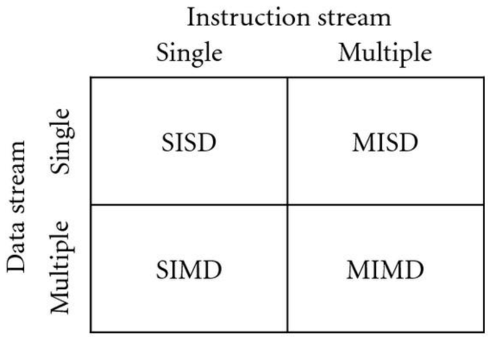
The goal of Widening is to invest the available parallel resources in a way that maximizes the search space exploration and, by that, the solution quality. It is critical to prevent the exploration of the same too similar solutions in parallel, so that the widened search is forced to investigate the search space more broadly, and increase the chances of discovering the global optimum. That is why we explore methods of *diversity*.

Moreover, the objective to avoid undesired overhead and the need for synchronization, which arises from the communication between parallel workers, leads us to look for *communication-less* Widening approaches, where the parallel workers do not communicate when selecting diverse paths through the solution space. Achieving a diverse traversal of the solution space is not difficult when the parallel workers are allowed to communicate, however achieving diversity in a communication-less setting, in which the parallel workers do not share information, is a more difficult problem.

This chapter is structured as follows. First, we will discuss the basic notions of parallelism and then will proceed with parallel data mining. We will then formally define what is *Widening* and will introduce the most basic properties of Widening. We will finish by introducing the structure of this dissertation, its main goals, and results.

1.2 Basics of Parallel Computer Architectures and Moore's Law

The book [123] was used for this section. The growing abundance of computing resources, coupled with the lack of improvement in single CPU performance has lead to the rapid rise of development of parallel algorithms. Parallel hardware and software are utilized nowadays to maintain the increase in processing power stipulated by the Moore's law



		Instruction stream	
		Single	Multiple
Data stream	Single	SISD	MISD
	Multiple	SIMD	MIMD

Figure 1.1: Flynn's taxonomy, presented in [54]. Image taken from [71].

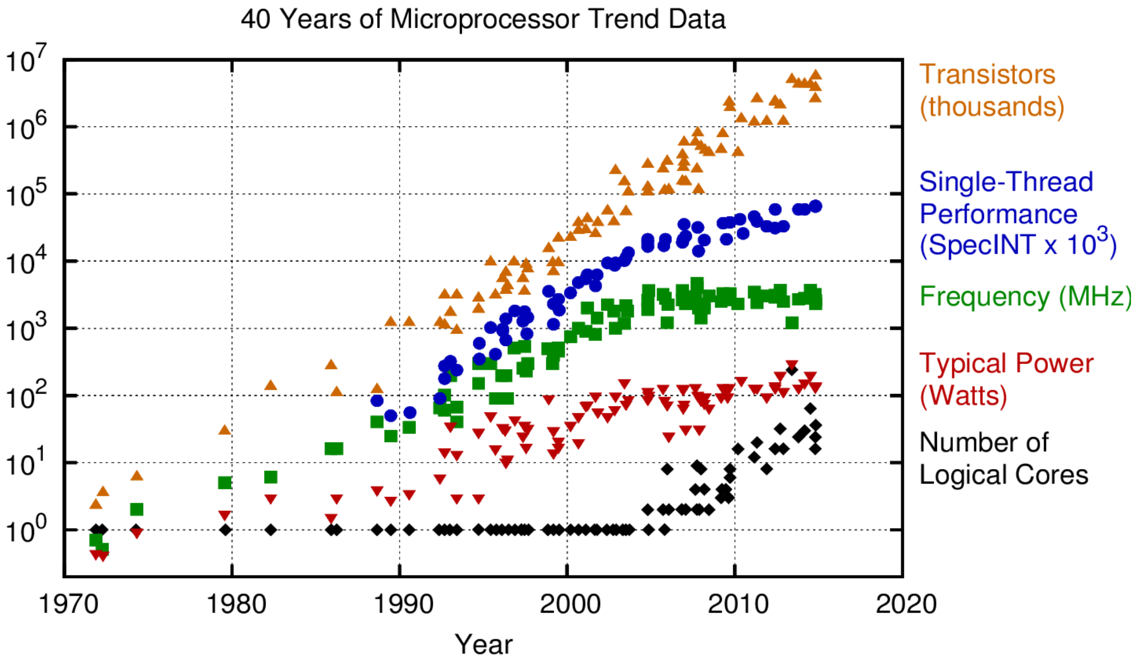
and a variety of parallel chip architectures with differing design and performance features exist.

Moore's Law

Moore's law declares that the number of transistors of a conventional processor chip doubles every 18–24 months. This consideration, first inferred by Gordon Moore in 1965, was accurate until the early 2000s. It was used as a benchmark for the research and development in the field and numerous innovations, such as the integrated circuit or DRAM, were establish in order for the technology to keep up. The most decisive factor for the boost of performance was the rise of clock speed. Different novelties like multiple functional units per processor were also used to improve the computational speed. However, these innovations to speed up single chips were exhausted by the early 2000s, and the industry started depending on multicore parallelism in order to improve processing power. The employment of many execution cores on a chip has lead to the decrease of execution time and the enhancement of computational accuracy.

1.2.1 Flynn's Taxonomy of Parallel Architectures

Flynn's taxonomy, shown in Figure 1.1, categorizes parallel computers in four classes dependent on the parallelism of data and instruction streams. The first is the classic sequential computer, Single-Instruction, Single-Data (SISD), in which a process executes one instruction using one data storage. Next, there is the Multiple-Instruction, Single-Data (MISD) paradigm, where multiple processing units execute different instructions at a time on common data obtained from a shared memory. This execution model is very restrictive and is not suitable for commercial use. Another model is the Single-Instruction, Multiple-Data (SIMD), in which the same instruction is performed at a given



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2015 by K. Rupp

Figure 1.2: "Current Trends in Computer Architecture. Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten New plot and data collected for 2010-2015 by K. Rupp" Caption and image taken from https://www.karlrupp.net/2015/06/40-years-of-microprocessor-trend-data/. Under Creative Commons Attribution 4.0 International Public License.

step by each processing unit on different data, stored in private data memory. It is suitable for tasks which demand high rate of data parallelism. In the Multiple-Instruction, Multiple-Data (MIMD) model each processing unit is working asynchronously executing independently its own instruction and using separate data. This model is the most prevalent and common-spread for everyday use.

Many architecture alternatives for multi-core processors exist, which are dissimilar in the number of cores, access to and size of caches, the use of heterogeneous components and many other possible variant features. We will not describe them in detail, because Widening is an architecture-independent approach, and the specific characteristics and design choices are not in the scope of this dissertation. What is important for Widening is the aforementioned trend in computation development towards an ever increasing number of compuational resources available. The existing and the potential future abundance of computing resources can be harnessed to improve the search space exploration and the model quality.

1.3 Parallel Data Mining

The following is based on [150, 149]. The growing size of data jointly with the trends for ever increasing parallel compute resources, has directed the field of data mining into designing parallel, more efficient algorithms. There are many challenges and important aspects for the parallelization of data mining algorithms. These low-level details of parallelization are not in the scope of this dissertation, which abstracts away such details and instead focuses on the behavior of the parallel workers and search space exploration. Still these implementation aspects are also important. For more detail, refer to [149, 150]. When trying to design efficient parallel data mining algorithms different potential problems are considered, for example, the synchronization between threads and minimizing of interprocessor communication, the strategy to split the work into small easily parallelizable tasks, also referred to as data decomposition. Another important challenge is the distribution of work across compute resources, known as load balancing. Depending on the architecture, different facets such as the type of parallelism of a load balancing strategy, can be critical. For example, data decomposition is relevant only for distributed memory approaches, and not for shared memory ones. Another possible consideration is the type of load balancing. In static load balancing, the work is apportioned to the processors prior to the onset of the task. Dynamic load balancing reassigns work from processing units with a heavy work load to ones with a smaller burden in order to amend disproportional work load. This approach is more costly and is suitable for work volumes, which alter over time.

1.3.1 Complete Search and Heuristic Search

The following is based on [149]. A complete search, which generates and checks all data-compatible candidates and finds the optimal solution, is possible only for a search space of a small size. However, in real-life problems, the search space is usually very large and a complete search is not possible and a heuristic search is adopted instead. By examining only a restricted number from all possible choices at a given step, the heuristic generation forgoes completeness as a result of trying to improve the speed. Generally, the more complex the model, the more a heuristic or greedy search is needed. The multitude of heuristics, developed in past decades make complex and immense search spaces possible to explore. The improvement in the efficiency is due to a limited exploration of the search space and has consequences for the quality of the discovered solutions. Often these heuristics resemble greedy algorithms and are guided by local information, or choose a locally optimal model at each step, which does not guarantee discovering of a globally optimal solution. The restricted investigation of the search space, which happens when a heuristic is used is the motivation behind Widening – the aim is to reduce the influence of the heuristic used, resulting in an expanded search space exploration and produce the same type of model but at a substantially improved level of accuracy. Widening achieves this by loosening the restriction on a given heuristic to pursue only one solution at each step. A *widened* heuristic considers several solution candidates in parallel. The most straightforward variant would be in a simple beam search as described in [7]. Widening differs from other approaches of investing parallel resources into improving model accuracy, such as ensemble methods. Our goal is to find *single*, *interpretable* models with *better accuracy*. For a detailed description of how Widening differs from existing approaches of parallel data mining, refer to Chapter 2.

Below we will describe the main idea behind Widening and we will define and formalize the most important aspects.

1.4 General Widening of a Greedy Heuristic.

This section is adapted from [86]

We can view many of the data mining algorithms as a *greedy search* through a space of potential solutions, the model *search space*. This search space consists of model candidates, from which the greedy algorithm chooses a locally optimal solution at each step, until a sufficiently good solution is found, based on some stopping criteria. The greedy search can, therefore, be schematically presented as an iterative application of two operators: *refinement* r and *selection* s.

During the refinement operation, a temporary model m is made more specific to generate new, potentially better, models (which we refer to as *refinements*). The *selection* operator chooses the locally best model from all possible refinements. For the purpose of this work we will assume the existence of a family of models $\mathcal{M}$, that constitutes the domain of the two operators. The refinement operator is model and algorithm specific and the selection operator is usually driven by the training data. We will investigate the selection operator in more detail, as it will be the tool we use to widen a greedy heuristic. It is usually based on a given quality measure ψ, which evaluates the quality of a model m from a family of models $\mathcal{M}$ (and therefore also its refinements):

$$\psi : \mathcal{M} \to \mathbb{R}.$$

Employing this notation, we can present one iterative step of the greedy search as follows:

$$m' = s_{\text{best}}(r(m)),$$

where

$$s_{\text{best}}(M) = \underset{m'' \in M}{\arg\max} \left\{ \psi(m'') \right\},$$

that is, the model from the subset $M \subseteq \mathcal{M}$ which is ranked highest by the quality measure is chosen at each step. Figure 1.3 depicts this view of a greedy model searching algorithm.

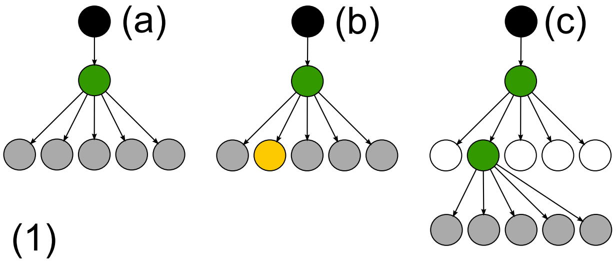


Figure 1.3: The classic heuristic (often greedy) search algorithm. On the left (a), the current model m is depicted in green, the refinement options $r(m)$ are shown gray. The selection operator s picks the yellow refinement (b) and the next level then continues the search based on this choice.

We can now also specify how we got to a certain model and define the concept of *selection path*, which defines how a specific model is reached:

$$p_s(m) = \{m^{(1)}, m^{(2)}, \dots, m^{(n)}\},$$

where the order is specified via the refinement/selection steps, that is

$$\forall i = 1, \dots, n-1 : m^{(i+1)} = s(r(m^{(i)})).$$

and $m^{(n)} = m$ and $m^{(1)}$ is a base model for which no other model exists that it is a refinement of. Note that the selection path depends heavily on the chosen selection operator s.

1.4.1 Widening of a Greedy Heuristic

In order to improve the accuracy of the greedy algorithm one has to deal with its inherent flaw – the fact that a locally optimal choice may in fact not lead us towards the globally optimal solution. To address this issue, we can explore several options in parallel – which is precisely what Widening is all about. How those parallel solution candidates are picked is the interesting question, which we will address later, but let us first look into widening itself in a bit more detail. Using the notation introduced above, one iteration of Widening can be represented as follows:

$$M' = \{m'_1, \dots, m'_k\} = s_{\text{widened}}\left(\bigcup_{m \in M} r(m)\right).$$

That is, at each step, the *widened* selection operator s_{widened} considers the refinements of a set M of original models and returns a new set M' of k refined models for further

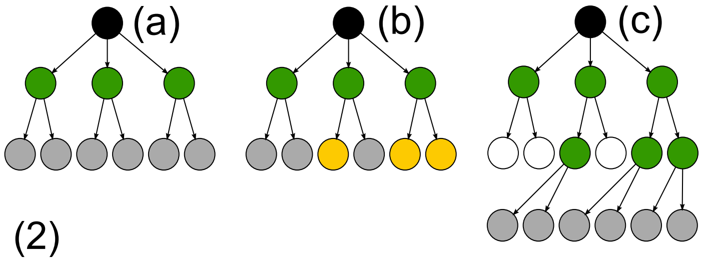


Figure 1.4: Widening. From a set of models M (green circles), the refinement operator creates several sets of models (gray), shown on the left (a). The selection now picks a subset of the refined models (yellow circles in (b)) and the search continues from these on the right (c).

investigation. We will refer to parameter k as the *width* of the widened search. Intuitively, it is clear that the larger the width, i.e. the more models (and hence selection paths) are explored in the solution space, the higher our chances are of finding a better model in comparison to the normal greedy search. Figure 1.4 illustrates this process.

An easy implementation of the above (what we will later refer to as $Top-k$ Widening) is a beam search. Instead of following one greedy path, the path of k best solution candidates is explored. However, this does not guarantee that we are indeed exploring alternative models – on the contrary, it is highly likely that we are exploring only closely related variations of the locally best model. In the area of genetic algorithms this effect is known as *exploitation*, that is, we are essentially fine tuning a model in the vicinity of an (often local) optimum.

In [7] we already described this approach to Widening. I In each iteration of $Top-k$ Widening each parallel worker selects the top k choices for the refinements of its model and from the resulting k^2 choices, the top k are chosen:

$$\{m'_1, \ldots, m'_k\} = s_{Top-k}\left(\bigcup_{i=1,\ldots,k} s_{Top-k}\left(r(m_i)\right)\right)$$

where s_{Top-k} selects the top k models from a set of models according to the given quality measure ψ. Obviously, $s_{Top-1} = s_{\text{best}}$.

In [7] we demonstrate that $Top-k$ Widening leads to an improved quality, with larger *width* k leading to better accuracy. However, two main flaws exist. The first problem, as mentioned above already, is that possibly only a small neighborhood of the best solutions is explored. Secondly, continuous communication is required between threads which contradicts our goal of wanting to keep the time constant.

Figure 1.5 illustrates the potential benefits of diversity in the search for the optimal model.

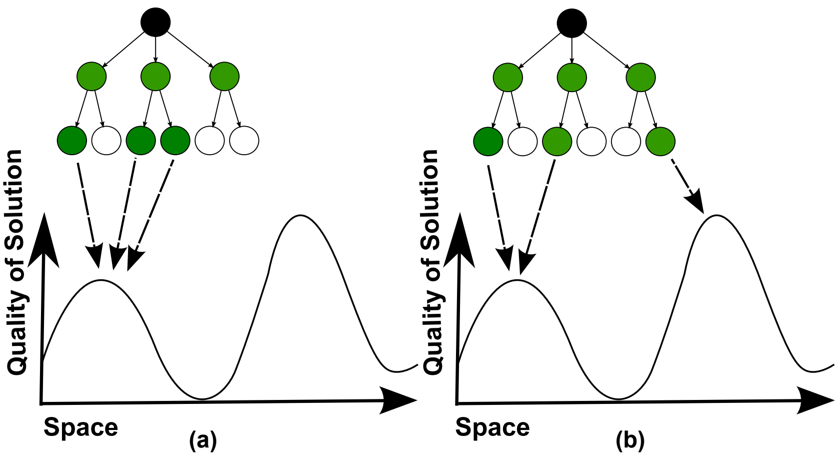


Figure 1.5: Normal Widening may lead to local exploitation only (a). Adding diversity constraints encourages broader exploration of the model space (b).

1.4.2 Diverse $Top - k$ Widening.

As discussed above we can tackle the first flaw of $Top-k$ Widening by enforcing diversity. One simple way to add diversity can be achieved by using a fixed diversity threshold θ, a distance function δ, and by modifying the selection operator $s_{Top-k,\delta}$ to iteratively pick the best k refinements, that satisfy the given diversity threshold. This can be summarized as follows:

```
$M_{\text{all}} = \cup_{i=1,\dots,k} r(m_i)$                create set of all possible refinements
$m_1 = \arg\max_{m \in M_{\text{all}}} \{\psi(m)\}$       pick the locally optimal model as first model
$M_1 = \{m_1\}$                                          add as initial model to solution
for $i = 2, \dots, k$:                                    iteratively pick next, sufficiently diverse model:
  $m_i = \arg\max_{m \in M_{\text{all}}} \{\psi(m) \mid \neg\exists m' \in M_{i-1} : \delta(m, m') < \theta\}$
  $M_i = M_{i-1} \cup \{m_i\}$
endfor
return $M_k$
```

This is a known approach for diverse subset picking, however, our second problem persists: we still require frequent communication among our parallel workers to make sure we pick a diverse solution subset among all intermediate solutions at each iteration.

1.4.3 Communication Between Parallel Workers and Diverse Widening

The simple beam-search approach $Top-k$ as presented requires communication between the parallel workers. The additional requirement of diversity between the selected models at each step will escalate this demand. This will contradict the second objective of our Widening approach – i.e., keeping the running time constant with respect to that of the original heuristic. To accomplish this we need to perform Widening in a communication-less fashion – where each parallel worker performs a part of the search, without communication or sharing of information with other parallel workers to be necessary. Enforcing diversity without continuously comparing intermediate models is more difficult. We can define individual quality measures ψ_i, by enforcing different preferences for different subsets. Due to the fact that we have no prior knowledge of the search space of models, defining which candidate solutions will be explored by which parallel worker in such a way that parallel workers explore the search space of models without repetition and in a structured, intelligent way is not a trivial task, as we will explain in Chapter 3. Especially, it is non-trivial to introduce diversity in the searches of the parallel workers, in a way, which does not require communication. This requires a mechanism to a priori assign candidates to a given worker. Such communication-less strategies for Widening is the focus of this dissertation.

1.5 Contributions

In this section, we will outline the main contributions of this dissertation, the different communication-less strategies for intelligent search space exploration, *communication-less* Widening. These strategies lead to the improvement of the quality of solution obtained by a data mining heuristic while avoiding the running time increase caused by communication between parallel workers. Widening was first introduced in [7] as a paradigm for an intelligent investment of parallel compute resources with the goal of improving the quality of solution obtained by data mining heuristics. My personal contribution was applying the paradigm of Widening for the greedy algorithm for the set cover problem. We motivate the need for communication-less Widening and describe the need for diversity for improved exploration of the search space as done in [86]. In this work, my contribution is to develop and apply simple diversity strategies for Widening of the greedy algorithm for SCP, which do not require communication and contrast them with diversity strategies, which require communication between the parallel workers. These communication-less methods perform very well in practice both with respect to model quality as well as efficiency.

The central focus of this dissertation is the communication-less strategies for structured search space exploration. We define neighborhood-based methods with different properties, introducing Widening via optimality, similarity and diverse neighborhoods. We discuss different methods for building diverse neighborhoods. We proceed to prove and discuss the theoretical properties of these neighborhood-based approaches in order to be able to compare them to the approaches which use communication. We define the concept of a *refinement graph* to represent the structured exploration of the search space. Different types of refinement operators lead to different types of refinement graph structures. For the simplest type of a refinement operator, (we call it refinement operator of type 1), the refinement graph is a lattice. Theoretical properties of Widening via optimality neighborhoods were analysed. This is accepted for publication in [89]. We analyse the theoretical properties for Widening via similarity neighborhoods and its use for exploitation. We introduce approaches for search space partitioning, which take advantage of the lattice structure of the space of models. We describe global diversity approaches, based on the lattice structure of the search space. We then apply all of these Widening approaches to the greedy algorithm for the set covering problem and the CN2 algorithm for the rule induction and evaluate how successful each approach is. Some of these results are already published in [87](accepted for publication), where communication-less strategies, including neighborhood-based approaches, for the Widening of the CN2 algorithm, were presented,[88](accepted for publication), where the greedy algorithm for the set cover problem was widened using different neighborhood-based approaches, and [89](accepted for publication), where theoretical properties of Widening via neighborhoods were investigated.

1.6 Structure of the Dissertation

This dissertation is structured as follows. First, it describes the existent research in parallel data mining and positions the Widening approach in that context with its goals and merits, explaining the novel contribution and importance of Widening, see Chapter 2. Second, in Chapter 3 we describe a formal framework, which defines *Ideal Widening*, we formulate the goal of the method and the theoretical properties needed in order for the Widening to achieve its goals. We define several formal Widening approaches in a communication-less scenario. These ideal approaches require explicit partitioning of the search space among the parallel workers. The main goal of Widening is the structured and full exploration of the search space. For some refinement operators and algorithms the graph representing the search space, defined by the refinement operator is known in advance (for some algorithms it is a lattice). However, the explicit structure of the search space is not always easily predicted, which is why any of the ideal methods are difficult to implement in the general case. Therefore, we come up with strategies, how to implicitly approximate the partitioning, by encoding a priori the desired behavior into the selection or refinement operators of the parallel workers. We describe a naive approach based on the simple assignment of different preferences. This approach is unstructured, with no guarantees, and requires a data-dependent parameter to be tuned, yet seems to work very well in practice. We proceed with a communication-less implicit approach, called *Widening via neighborhoods*, which explores the search space by using neighborhoods of models in Chapter 4. We investigate the theoretical properties of this approach in Chapter 5, which show under certain assumptions, how similarity and optimality neighborhoods can be used for better *exploration* and *exploitation* of the search space. In Chapter 6 we describe Widening approaches, which use *global* approaches to diversity and the knowledge of the lattice structure of the search space. Then we proceed with practical applications and demonstrations of Widening, namely Widening for the greedy algorithm for the set cover problem in Chapter 7, and Widening of the CN2 algorithm for rule induction in Chapter 8. A conclusion, which summarizes the main contributions, as well as some open questions, is presented in Chapter 9. There we provide an outlook and outline potential future focus for research.

Chapter 2

Related Work

Parts of this chapter is adapted from [86].

2.1 Parallel Data Mining

A wealth of related work exists around the parallelization of data mining algorithms, most of it dealing with speeding up sequential algorithms. Widening differs from other approaches of investing parallel resources into improving model accuracy, such as ensemble methods. Our goal is to find single, interpretable models with better accuracy. In [8], the author already introduced the idea of improving quality by using more parallel resources, but he investigates an extensive area of applications, ranging from cryptography to game playing, while we are focused on data mining.

2.1.1 Criteria for Evaluation and Classification of Parallel Data Mining Algorithms and Platforms

In order to demonstrate the contribution of our work in the field of parallel Data Mining, we will assess how Widening strategies differ based on their goals and qualities from the existing research in the field using the following criteria:

- Size: What size of data is the algorithm designed to handle. Is the algorithm focused on handling big data or algorithms handling normal-sized data?
- Speed: Is the parallelization focused on improving speed?
- Accuracy: Is the parallelization focused on enhancing the accuracy of a sequential data mining algorithm?
- Interpretability: Is the model obtained by the algorithm interpretable or not?

- Flexibility: How flexible is a given parallel approach with respect to applying it to different types of sequential algorithms and existing technologies?
- Search space coverage: Does the parallelization approach improve search space exploration?
- Distributed (heterogeneous) versus non-distributed (and homogeneous) data: Is the approach suited for heterogeneous data or is its focus homogeneous data?

The following are the merits and characteristics of the Widening approach:

- It focuses on improving accuracy, instead of efficiency.
- Our approach results in the same single simple model as does the original greedy algorithm, only more accurate.
- It is not focused on big data problems, instead improves the accuracy for normal-sized problems.
- The time necessary to obtain the enhanced model is known. It is the same (or close to that) of the initial heuristic.
- Enhanced search exploration. Widening is focused on a structured intelligent exploration of the search space. Widening strategies are focused on investing parallel resources in improving the quality of the solution via the enforced use of diversity.

Our approach is focused on dealing with normal-sized data problems, in a scenario where accuracy is more important than speed. We will demonstrate that based on the above criteria our contribution is novel and is not redundant with any other research done in the field of parallel data mining.

2.2 Speed-Up Through Parallelization: Algorithms Focused on Improving Efficiency

Most of the principal data mining algorithms for classification, association rule mining, and clustering have been parallelized with the goal of speed-up through the adoption of various approaches. Bellow, we will present only the most important parallelization strategies used for improving efficiency, because they are not related to Widening. For a detailed review in detail refer to the following surveys: [153, 148, 95].

The parallelization of decision tree induction has been broadly explored. SLIQ [114] is one of the oldest decision tree algorithms, which scales well. The vertical data format facilitates a pre-sorting of the attributes, which renders the repeated sorting at each

node unnecessary. Nevertheless, it uses a structure, stored in memory, which limits the algorithm's scalability. A decision tree algorithm called SPRINT [129] eliminates these constraints by avoiding the necessity for all or a part of the data to be stored permanently in memory and scales better over large datasets. It showed a very good speed-up on the IBM $SP2$ distributed-memory machine. As a consequence, many parallelization strategies of decision tree algorithms adopted SPRINT-like strategies, and apply data parallelism. One of the more prominent examples is Zaki et al. [151], where the SPRINT was parallelized for symmetric multiprocessing machines. Other decision tree parallelization strategies employ data task parallelism [38] or hybrid parallelism [132, 102].
Association rule mining is another notable and extremely well-studied data mining algorithm. A survey [147] outlines a multitude of parallel and distributed algorithms for association rule mining. A lot of the methods detailed in the survey are parallel variants of already existing sequential algorithms. The Apriori algorithm is accepted as the most fundamental sequential algorithm for association rules mining. It is a merge type of algorithm, which limits the full exploration by the adoption of constraint inclusion, referred to as "support". The Apriori method [3] is used by the large bulk of parallel association algorithms [131, 2, 152, 79].

Direct hashing and pruning (DHP) [119], is a sequential algorithm, which enhances the Apriori approach by introducing a hash table technique, which pre-computes approximate support of small itemsets in the beginning, and eliminates future infrequent candidates. A parallel algorithm based on DHP was developed by the same authors [120]. Approaches based on the sequential version of the Eclat algorithm were adopted in Max-Eclat, Clique, and MaxClique [154]. These algorithms exploit the structural properties of frequent itemsets to speed up the search but do not seek solution quality improvement. Parallelism in the clustering algorithms has been used for speedy clustering strategy as well as for the computing the distance in a fast manner. Partition clustering [92, 44, 93] algorithms most commonly rely on message passing interface to exchange information and occasionally characteristics of the network topology is additionally employed to improve the effectiveness of data sharing. In others, a master-slave configuration with a message-passing model is used to efficiently compute the similarity between data. Hierarchical clustering [57, 118] is more computationally demanding compared to other clustering approaches because it involves computing the level of clustering as well. Some single-linkage algorithms are parallelized using a hypercube network to reduce the computation of the minimum spanning trees.

While Widening is not focused on improving efficiency and instead invests parallel resources into improving quality of the solution, its goal is predictable running time, that of the sequential heuristic.

2.3 Flexibility

When we discuss "flexibility" we refer to two aspects. Firstly, how much a given parallelization is dependent on the technology it is developed for. For example, algorithms parallelized to speed-up using the map/reduce paradigm are very specific, paradigm and platform dependent and thus lack flexibility. Additionally, we refer to whether or not a given approach is applicable to many data mining algorithms. For example, not all algorithms can be parallelized well via the map/reduce paradigm. Favorable algorithms for map/reduce parallelization are those with few iterations and long inner-loop cycle – Naive Bayes, k nearest neighbor, k-means, and expectation-maximization. Algorithms with multiple iterations and short inner-loop cycles, such as AdaBoost, support vector machines, and logistic regression are not well suited for the map/reduce approach.

In contrast, our goal is to provide an approach that ignores low-level details and has the flexibility of the parallelization and is applicable for multiple (most) data mining heuristics. Still, Widening carries algorithm specificity with it. Certain heuristics will be more easily widened than others.

2.4 Model Quality Improvement

A number of papers also concentrate on improving the accuracy of the models. Some attempt to improve the greedy algorithm by making less greedy choices, others learn more models and aggregate them in different ways as ensembles or, as some parallel metaheuristics do, explore the search space in parallel in a randomized fashion.

2.4.1 Look Ahead Strategies

Similar to the aspect of *deepening* as discussed in [7], a number of other approaches exists, which try to reduce the effect of greedy, local optimum picking by taking into account how future decisions affect the overall performance. In [124, 115, 45, 48] a few such approaches for decision tree induction are described, which improve the split point choice by investigating how the split criterion behaves for the given choices considering a certain number of additional splits. This type of look-ahead strategy is very hard to parallelize in such a way as to ensure that overall computation time remains constant compared to the normal greedy method.

2.4.2 Interpretability and Ensemble Learning

Ensembles combine together models to achieve a better prediction accuracy than from any of the models when used separately. Among the most important examples are bootstrap aggregating or bagging [18], boosting [125], and random forests [19]. These techniques are suitable applicants for parallelization [146, 104, 37]. A survey of ensemble clustering methods, where the employment of metaheuristics induced enhanced clustering accuracy, is described in [140]. However, a high degree of accuracy comes at the price of interpretability as these methods do not result in a single interpretable model, which is contrary to the goal of widened data mining.

The interpretability of other parallel data mining algorithms is generally not affected. As we already stated, the Widening of a given algorithm results in an interpretable single model, as does the original sequential heuristic that it was applied to.

2.5 Greedy Search Algorithm Improvement

There is a wealth of literature focusing on the improvement of greedy search algorithms in general, for example, beam search-like algorithms. These improvements, however, do not promote the complete exploration of the model search space. The look-ahead strategy, mentioned above, has been utilized as an enhancement for the greedy heuristics in general as well [124]. In [130], an approach is presented for incorporating diversity within the cost function used to select intermediate solutions. In [80], the authors use the observation that, in most cases, failing to find the optimal solution can be explained by a small number of erroneous decisions along the current path. Consequently, the enhanced search for a fixed depth-first explores the left-most child as suggested by the original heuristic and, if no solution is located, it continues with the nodes, which are dissimilar by just one point from the greedy choice and so forth. The Widening proposed here performs a similar search for alternatives but in parallel. In [50], adding diversity to a simple k-best the first search was shown empirically to be superior to the greedy search heuristic.

2.6 Size of Data: Algorithms for Big Data versus Algorithms for Normal-Sized Data

2.6.1 Specific Frameworks (MapReduce)

MapReduce [41] is the most well-known programming model for executing tasks on big data sets in a parallel and distributed fashion on a cluster or a grid. It is based on a decomposition of the algorithm into a map and a reduce step. Because of its ingrained properties, MapReduce can only be applied to specially designed variants of the data mining algorithms. For example, [155] presents parallel k-means clustering and [142] proposes the scaling of genetic algorithms using MapReduce. Many other papers focus on single MapReduce implementations of other data mining algorithms. Chu et al. [26] present a more general approach to parallelize algorithms by using a summation representation of the algorithms; it is applied to locally weighted linear regression, k-means, Naive Bayes, support vector machines, expectation-maximization, and others.

Despite its potential for scalability, MapReduce has inherent flaws – it is not designed to deal well with moderate-size data with complex dependencies; it is not appropriate for algorithms that are iteration-based and, especially, communication between the parallel workers (see [109] for a more detailed discussion of these issues). So while MapReduce allows enormous amounts of data to be processed, it is not a general framework for the parallelization of complex algorithms. Having said that, MapReduce may well be a suitable framework for the widened data mining algorithms described here – however, published work has so far focused on larger data and the potential for creating better models based on this increased ability to process data. The focus of widened data mining is to create better models from the same amount of data via a smarter exploration of the search space. GPUs are successfully applied for the speed-up of Big Data algorithms. For example, a map-reduce model for GPUs is presented in[81, 49, 135, 85, 46].

2.7 Distributed Data Mining

Nowadays, centralized storage of data is typically costly and unrealistic, more vulnerable to security risks, and limits scalability. In today's world, very often, the data is stored in a distributed form.

Distributed data mining (DDM) is concentrated on developing methods and algorithms for performing data analysis in a situation in which the data and the computational resources may be scattered and stored in different locations. In order to improve the efficiency and scalability of the algorithms in such a setting, DDM turns to parallelism. The algorithm is employed to each data site independently and concurrently, resulting in one local model for each site, built only on local information. Upon completion, all local models are combined into a global model. To increase the global knowledge of the local models, some data transfer is required.

Different types of DDM environments can be divided into, i.e. homogeneous sites with shared characteristic and sites, which adopt different characteristics.

2.7.1 DDM from Homogeneous Sites

The algorithms used for mining homogeneous sites are usually parallel by design; different classifiers are learned using local data and then aggregated using meta-learning approaches. A multitude of meta-learning strategies can be utilized such as bagging [18], stacking [145] and others.

2.7.2 DDM from Heterogeneous Sites

Learning in parallel from heterogeneous data sites logically results in models, which are different. It is difficult to learn globally representative models and to correctly aggregate the local models, achieved by learning from heterogeneous sites in parallel. Of course, minimal communication is desirable in highly distributed settings. Meta-learners, which use order statistics to combine the models resulting from multiple learning agents, which are trained on different distributed data sites, are especially well suited for aggregation of heterogeneous models [139].

In [94], the authors propose an approach, called collective data mining, which aims to improve the accuracy of locally learned models and their aggregations, with minimal communication, by using orthonormal representations of functions. Collective data mining algorithm approaches include collective PCA [97], clustering [96], Bayesian learning [25], collective multivariate regression [83], and others.

Papyrus [12], another platform for distributed data mining in a heterogeneous setting, learns intermediate models locally for each cluster and moves these locally learned predictive models to a central location, where a multitude of aggregation methods are used to determine the final model.

JAM [133] is originally a fraud detection framework. Each learner employes a different algorithm to build a model on different data sites then the resulting set of models are aggregated using standard approaches like ensemble learning, or field-specific ones like *knowledge probing*, to form a meta-classifier with an enhanced predictive accuracy. Knowledge probing can result into a descriptive and interpretable aggregate model.

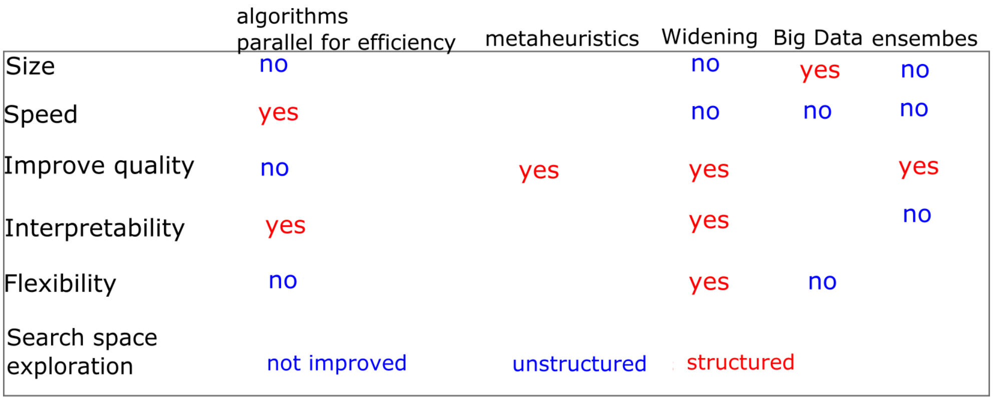
2.7.3 Widening and Distributed Data Mining

These DDM approaches have intrinsic parallelism by nature, but they are focused on data partitioning and model aggregation, which rarely results in an interpretable model. While some of these are also applicable for Widening, the approaches above are specialized to solve challenges of data mining in a distributed context. Widening is not specialized in distributed or heterogeneous data in particular. A common theme between Widening and DDM is minimizing the communication between the parallel workers in the case of Widening, and between the decentralized units in the case of DDM.

For visual representation of how Widening fits into the context of existing parallel data mining research, refer to Figure 2.1.

2.8 Population-based Metaheuristics

Even though due to their approximate and non-deterministic nature, metaheuristic searches offer no guarantees for finding globally optimal solutions, they have been proven



	algorithms parallel for efficiency	metaheuristics	Widening	Big Data	ensembes
Size	no		no	yes	no
Speed	yes		no	no	no
Improve quality	no	yes	yes		yes
Interpretability	yes		yes		no
Flexibility	no		yes	no	
Search space exploration	not improved	unstructured	structured		

Figure 2.1: A comparison between the most important characteristics of Widening and other data mining approaches.

extremely useful for solving large problems of high complexity. Metaheuristic strategies have been categorized as either *population-based* or *neighborhood-based* approaches. We will first discuss parallel and sequential population-based metaheuristics. The next section is dedicated to the neighborhood-based metaheuristics because they have some similarity to Widening.

2.8.1 Parallel Population-based Metaheuristics

Strategies based on stochastic learning algorithms, such as genetic algorithms, are naturally parallelizable. Their parallelization can be achieved by a straightforward execution in parallel of independent copies of the same algorithm (only parametrized differently). In the end, the best solution is chosen from the ones obtained by these independent searches. This simple approach to parallelism achieves an improvement in model accuracy [137]. Examples of using (parallel) genetic algorithms for data mining include GA-MINER [53], REGAL [63], and G-NET [62].

The main difference between Widening and these metaheuristic methods is that the former is focused on exploring the search space in a *structured way* as opposed to the randomized nature of these other methods.

2.9 Neighborhood-based Metaheuristics

This section is based on [13], [60], [60, 70]. In Widening, we use neighborhood-based approaches not in order to find a local optimum in a particular neighborhood, but with the idea of achieving global search space exploration and to, given enough resources, explore fully the search space, as described in Chapter 3.

Many known approaches for solving optimization problems apply the idea of *neighborhoods* to perform a search. These methods move repeatedly from one solution to a neighboring one, which is evaluated as the best possible neighbor, according to some criteria. Below we briefly present the most important neighborhood approaches.

2.9.1 Local Search

This section is based on [13]. The local search begins with some initial solution and reiterates minor modifications in order to improve it. The approach necessitates the concept of a neighborhood to be defined in the search space. A modified solution neighbor is selected as the new temporary solution only if it performs better than the already chosen temporary solution according to a preselected evaluation function and, if not, a different small alteration is checked and evaluated. An already selected model x with a neighborhood $N(x)$ is substituted by a neighbor $x' \in N(x)$ if x' is evaluated as better than x . This step is performed until a solution is discovered, which is not worse than all the models in its neighborhood.

2.9.2 Simulated Annealing

This section is based on [13]. Simulated Annealing [100] is a metaheuristic approach for the approximation of the global optimum for complex, most commonly discrete, optimization problems. Simulated annealing fares better when global or a close approximation of the global optimum is needed, compared to other methods, which easily converge to local optima. The method has borrowed ideas from a process in metallurgy with the same name, which alters the physical properties of a solid by employing heating followed by controlled cooling down. This algorithm is akin to the simple local search, but it prevents potential converging of the search to local optima by allowing so-called "jumps", which consist of choosing models of worse quality based on some preselected probability.

2.9.3 Tabu Search

The tabu search [64, 67] enhances the local search, by encoding a propensity towards unvisited or promising sectors of the search space, which have shown to have high-quality models. The search uses tabu lists and different types of memory (long and short) to avoid revisiting regions unless they contain promising solutions. As a result, the search achieves a diverse exploration by picking previously unexplored regions, while also using exploitation for regions of the search space, which are known to contain promising solutions. The essential features of tabu search are detailed in [64, 65, 66, 103, 67].

2.9.4 Large Neighborhood Search (LNS)

Efficiency is the rationale behind the use of small neighborhoods by most local search strategies. Even though large neighborhoods facilitate the discovery of higher quality solutions, they are more costly to explore at each iteration, because of their size. Due to the higher cost, fewer search executions can be performed to induce an inferior final result. Large-neighborhood search methods [4] tackle this issue by adopting suitable heuristics which facilitate the efficient investigation of neighborhood structures of high complexity. The very large scale neighborhood (VLSN) search approaches employ strategies, which do not necessitate a complete enumeration of the neighborhood but instead use approximate evaluation approaches. An important illustration of the VLNS presented in [108], where it is applied to enhance a well-known algorithm for the traveling salesman problem (TSP). An example of VLNS based network flow is presented in [138]. An example of applying VLNS for the solution of special cases of a hard optimization problem is described in [69].

The existing parallelization approaches for LNS and VLSN are only focused on improving the speed of processing of the large neighborhoods.

2.9.5 Greedy Adaptive Search Procedure (GRASP)

GRASP integrates together a greedy search and a neighborhood search. Each iterative step involves the use of randomized greedy construction of multiple starting solutions and local neighborhood search processes, which exploit each of the starting solutions, [51].

2.9.6 Variable Neighborhood Search (VNS)

The VNS search starts at a local optimum and then constantly assesses expanding neighborhoods of that solution until a solution of higher quality is discovered. This process is repeated with the newly discovered solution as a center for the exploration of neighborhoods or broader and broader size. This strategy utilizes the consideration that for a multitude of problem types, local optima are adjacent to the global optimum and that the global optimum is an optimum within each neighborhood of a search space.

Among the benefits of this search is that few if any parameters need to be set, which leads to the uncomplicated and straightforward discovery of high-quality solutions. It is also flexible, and to make the search more efficient different heuristics can be utilized for the local search. VNS-based algorithms have been employed to solve various optimization problems, both combinatorial and continuous, as well as clustering and others.

2.9.7 Parallel Variants of Neighborhood-based Metaheuristics

This section is based on [141] and the state-of-the-art surveys [34, 55], as well as the book [9], which present the major results in the field of parallel metaheuristics. In order to discuss the parallelization of neighborhood-based metaheuristics, or parallel local search, the authors of the surveys and book cited above, develop taxonomies of the different parallelization types. These taxonomies, while not identical, are very similar, especially with respect to the parallelization approach that concerns us – it is referred to as either parallel multiple walk without communication or as multiple independent runs in the different literature reviewed below. It refers to a parallel approach, based on multiple search space explorations in parallel, in which the parallel workers do not exchange information.

In [141], a survey of parallel local searches is presented. It considers different types of local search algorithms and presents existing approaches to parallelizing them. The survey classifies the algorithms in two main classes – parallel single walk and parallel multiple walk. In the single walk algorithms, one or many steps of the single walk are performed in parallel. At each step, the neighborhood is distributed among parallel workers, with the goal of improving the efficiency of the algorithm, which is not related to the goals of Widening . We aim to split the entire workspace via Widening, while avoiding communication between parallel workers. Our goal is escaping local optima and thus alleviating the problem with data mining heuristics. In contrast, the distributed neighborhood that is investigated by parallel single walk algorithms is the same as a traditional neighborhood, and the solution discovered is the same as that obtained by the original heuristic, only faster.

The concept of multiple walks as a way to parallelize local search is closer to the notion of Widening. In this approach, there are multiple simultaneous walks through the search space. The multiple walks that are discussed in this work are with and without communication. For the multiple independent walks, the parallelization is achieved by simply starting the algorithms multiple times. The different paths of the walks are a result of diverse parameter values or starting points. While this does produce a higher probability of finding a good solution, this approach is very simplistic and does not take into account the goal of exploring the entire search space. The interactive approaches are similar to the ones with diversity measures that require communication and this results in greater overhead and computational costs with the increase in parallelism.

In contemporary state-of-the-art surveys, the field has shifted in the direction of the so-called multiple walk parallelizations of metaheuristics. In [34], the goal of improving the solution quality as a result of parallel multi-start heuristics is explicitly stated and reported in several parallelization cases.

Parallel metaheuristics allow for both, an improvement in efficiency as well as in solution quality. In [9], parallel metaheuristics are viewed as a separate class of heuristics altogether. The authors provide a similar taxonomy and present a more detailed study of the field. Often the parallel variant returns a better solution than the sequential heuristic, which makes the analysis of the parallel metaheuristic methods more complicated. This is why the authors suggest that the evaluation criteria of parallel metaheuristics should incorporate the solution quality as well as the speed up measure. Three main types of parallelizations are recognized, similar but not identical to the types differentiated above. Below we will present these different types of parallel approaches and will focus mostly on *type 3* or multiple runs, especially multiple *independent* runs, because this type of parallelization is the most is relevant to Widening.

The type 1 parallelism of a local search is a low-level parallelism, parallel evaluation of the neighborhood at each step, equivalent to the single walk type, presented above. It results in the same solution as the original heuristic and is focused on speed-up.

The type 2 parallelization approach divides the variables across the parallel workers and as a consequence, each worker explores a restricted part of the search space. While this approach does explore solution paths different from those of the sequential heuristic, it still is different to Widening in terms of goals.

The type 3 parallel approaches are implemented as many parallel searches of the space of solutions. We will discuss in more detail the main features and the key research done with type 3 parallelism, as it has elements, similar to Widening, and we want to show the similarities and stress the differences between the Widening approach and these methods. Furthermore, some elements can be applied to the Widening approach as well. Just like Widening, one of the goals of the multi-run parallelization is to achieve a better quality of the solution.

Parallel multi-start heuristics can be categorized into multiple independent runs, where the parallel workers do not share information, and cooperative approaches, which take advantage of information exchanged between the parallel workers. The cooperative metaheuristics can exchange information in a synchronized fashion or have asynchronous communication.

Cooperative Multistart Runs

The approaches based on multiple cooperative walks represent the greatest advance in this field. These approaches bear similarity with some of the aspects of Widening. The goal of obtaining a better solution quality through a more thorough search space exploration is explicitly stated by the authors. Typically, the quality of the solutions discovered by these methods surpasses that of the serial methods. However, the necessity for constant and continuous exchange of information in the synchronous cooperative approaches results in worse running time and increased overhead. The asynchronous co-operating strategies were designed especially to overcome the overhead problems. Each individual parallel worker has a different starting point and an individualized search strategy and it communicates with the other parallel workers through a central memory without synchronization. For some approaches, called adaptive memory strategies, the central memory records partial solutions This central memory can store partial or complete solutions, while for the ones known as central memory approaches, the complete solutions are kept.

The strategies, based on adaptive memory have proven to be very useful in type 3 parallelizations of tabu search for vehicle dispatching problems and real-time routing [59]. It was also advantageous for the vehicle-routing problem with time windows, described in [11]. Another example of an adaptive approach, where the broadcasting of the fittest partial solutions, was used for the transfer of information between workers can be found in [128]. In [32], a tabu search, with information sharing was applied to the fixed cost capacitated multicommodity design problem. An example of a parallel cooperative multi-start of GRASP with adaptive load balancing is presented in [113]. A VNS-based cooperative asynchronous multi-start search for the p-median problem is published in [33].

Some examples of parallel synchronous cooperative search include [56], where multiple VNS are performed, and [121], where a synchronous tabu search is presented. For simulated annealing, synchronous and asynchronous cooperative approaches are described in [78]. Ref. [106] describes synchronous and asynchronous cooperative approaches based on Markov chains.

Hybridized Approaches of Multistart Runs

As the name suggests, hybridization metaheuristics incorporate multiple properties from several types of metaheuristic approaches in order to enhance the search and improve the model quality. This category of strategies has demonstrated the biggest improvement in terms of efficiency and solution quality compared to the other parallelization strategies. The hybridized approach integrates different heuristics and profits from their respective strengths. Below we will only briefly describe some examples of hybridization, for additional details and information, refer to the following state-of-the-art overviews and surveys [55, 35, 9].

Hybrid metaheuristics can be multiple independent runs (MIRs), which are relatively rare, or cooperative multi-start searches, regarded as state-of-the-art in the field, improving both efficiency and solution quality. Examples of hybrid MIRs include the GRASP MIR, where an intensification method called *path relinking* was used for the job scheduling problem [5], as well as a GRASP and path-relinking hybridization for the tree index assignment problem [6]. Path relinking and scatter search [68] are hybridization strategies, which take advantage of a long-term memory to direct towards sectors of the search space, which could contain the optimal solution. An example of path relinking, which relies on cooperation for its strategy, is shown in [31]. A parallel simulated annealing asynchronous hybridization is described in [112]. The authors of [30] present a hybrid approach, in which an asynchronous tabu search is utilized in order to build a set of "good" solutions and these then are used as a starting point for a genetic algorithm-based search. This approach was tried for the problem of multicommodity location-allocation. Two main advantages are shown of such a hybrid approach. Namely, the initial population of high quality, discovered by the tabu search contributes to the discovery of even better solutions, and the high-variety populations lead to diverse exploration. In [76] a record-to-record algorithmic approach [24, 107] was combined with the set-covering algorithm to determine new solutions and select the best ones, which enhanced both speed-up and quality of the solution.

Multiple Independent Runs (MIR)

MIR methods are not as popular as the multiple cooperative runs. These approaches are the ones that resemble the goals of Widening the most. We will discuss the difference between them and Widening in the subsection below.

Some significant examples of MIR are shown in [52], [56], where an independent multistart, which uses VNS is presented, tabu search [17, 10], and an independent parallel multistart [136].

GRASP is most commonly parallelized as MIR strategies, where each autonomous search uses the same sequential algorithm and identical data, but its own distinct seed to generate an individualized starting point of the search. In these approaches, the only prevention taken to avoid the workers investigating the same search paths is the adoption of dissimilar starting positions in the search space. These search techniques often use a master-slave configuration, where the master collects and stores globally the best temporary solutions, distributes data and generates seeds to be used by the slave parallel workers. The runs of the searches are autonomous with limited or no exchange of information between the parallel workers. Redistribution of workload over parallel workers may be used to tackle possible load balancing issues. This approach to multiple runs is oversimplified, and the diversity of exploration is due entirely to the distinct starting points. In later search strategies, multiple runs utilize previously discovered good solutions for seeds. In contrast, Widening aims to achieve a more detailed exploration of the solution space by using sophisticated strategies of search space partitioning, or at least diverse exploration without communication, instead of just using separate starting points as a source of diversity. We will outline the differences between Widening and the MIR approaches below.

2.9.8 Widening versus Parallel Neighborhood-based Metaheuristics

Local approaches typically search for a good enough solution within some neighborhood, namely, a *local optimum*. In contrast, even when using neighborhoods, the goal of the

Widening is, on the contrary, an all-encompassing exploration of the solution space, and discovering as good as possible, or ultimately, the globally optimal solution. In contrast to the Widening approaches discussed in this dissertation, the parallel local search algorithms presented have as the main goal to discover a sufficiently good solution in an efficient manner, while not necessarily looking at the parallelism as a way to find a better solution, or even trying to escape a local optimum.

We are interested in advanced strategies, which target a search space exploration in parallel, which is systematic and structured and passes through all the important regions of the search space, and which leads to the discovery of the global solution, without the need for communication between workers. The goal is to develop smart parallelization techniques, where each worker has an individualized behavior, that takes into consideration information of the search space, instead of performing the same linear algorithms in parallel with dissimilar starting points. We are interested in discovering exploration approaches, which, by modifications in the individual search strategies of each parallel worker, discover the optimal solutions in diverse parts of the search space, and ultimately, the global optimum.

While a lot of progress has been made in search space exploration, especially when it comes to the cooperative multiple walks, they boost the exploration via exchanging information or via adding randomization/genetic search hybridization approaches.

Even more sophisticated strategies, such as path relinking and scatter searching, while producing higher solution quality, do not use the parallelism for better exploration of the search space. They do consider diversified sets of solution candidates, but there is no explicit individualized strategy for each parallel worker, which would aim at search space partitioning. Instead, yet again, the same approach for searching is used by each parallel worker, however different starting solutions and randomization are used to achieve some individuality of the search path.

2.10 Parallel Deep Learning

2.10.1 Deep Learning

Deep learning captures and models complex hierarchies and dependencies hidden in data. Often it relies on different types of multi-layer artificial neural networks (for example, recurrent, convolutional or deep belief neural network), with many processing layers used for data transformation. Prominent examples of the diverse application of deep learning algorithms include [73, 105, 14, 127, 84].

2.10.2 Parallel Deep Learning

Almost every aspect of deep learning is intrinsically parallelizable. The main difficulty for deep network parallelization is the way the algorithm backpropagation functions. In backpropagation, it is a requirement that the calculation and update of a very large number of parameter values from one iteration must be finished before the start of the next one [43]. Due to the extreme computational costs related to training highly complex models, the relevance of the deep learning approaches is dependent on the computational processing innovations. Big work has been put into improving efficiency and speed-up of the most commonly used algorithms for deep learning. With the utilization of GPUs in place of the traditional CPUs, the methods gained significant speedup, which brought about important advances in various fields, for example, computer vision. Famous examples include [126, 36, 101]. Nevertheless, further research into the speed-up of deep learning algorithms is required, since, regardless of all the advances, the state-of-the-art methods are not fast enough for many real-world problems. Despite its unquestionable applicability, its high accuracy prediction, flexibility and great successes in different fields, deep learning does not offer a single interpretable model, and so far the existing parallel versions of deep learning systems are focused on speeding up the algorithms in order to handle the problems of the method's enormous computational costs. In contrast, Widening is focused on achieving an interpretable model of better model quality through a proper investment of parallel resources.

2.11 Topological Data Mining

This section is based on [20, 47, 21].

Topological data analysis (TDA) is employed to derive knowledge from complex data by examining its geometric characteristics and evaluating their statistical significance. The data is first presented in the form of points in a metric space and then converted to a topological structure to in order to study its shape and properties. The main underlying assumption is that shapes that persist, for various parameter values, are descriptive for the data. The most commonly mentioned topological visualization methodologies are the barcode [23] and the persistence diagram [28]. These methodologies have been shown to be useful for feature detection in data of high dimensionality. A main challenge in the field is that the integration between topological and machine learning methods is not straightforward. For detailed knowledge about the field, refer to the following surveys [61, 21, 47].

According to [20], evident advances of TDA approaches include its application to breast cancer [117], gene detection in microarray data [42], natural image analysis [22], sensor networks [40], orthodontics [82].

While not related to parallel data mining, topological data analysis uses formal generalized approaches to infer properties of the data. Widening also aims to formalize and generalize data mining approaches. Data with different topological properties may need different Widening strategies and the knowledge of topological structures may prove to be useful for Widening.

2.12 Conclusions

In this chapter, we discussed that Widening, as a type of strategies for investing parallel resources in a way that leads to better search space exploration and as a result, to discovering a better solution than with the original heuristic. The running time is also of importance. We would like the running time not to worsen, compared to that of the original heuristic. Widening results in an interpretable model.

Most of the parallelizations of existing heuristics are done with the idea of improving efficiency or are big data algorithms. Genetic algorithms, ensembles, and random forests are naturally parallelizable. They do induce a better model quality, however, they do not result in an interpretable model.

Optimization-related metaheuristics and approximation algorithms are more closely related to Widening. Different search strategies and their parallelizations were discussed. However, very few of the parallelization strategies aim for an improvement in the quality of the obtained model, the focus of most is on efficiency. The approaches, where the solution quality bettered, are simplistic, based on starting several identical parallel searches simultaneously in parallel, just with different starting points. They do not employ any sophisticated strategies for better (or ideally full) exploration of the search space of models. Nevertheless, the research done in the field of optimization has developed a vast number of sequential and parallel search strategies that can be applied to machine learning in general, and Widening in particular. The opposite is also true, machine learning approaches are heavily being used in the field of optimization, especially when evaluating the properties of the search space.

Chapter 3

Ideal Widening

The main portion of this chapter is directly quoted from [16], it does not represent the author's own work and is used in this dissertation, which describes the main goals for Widening and sets the context for the author's own research, and is important for completion and referencing. The final section 3.4 is based on [86] and is the author's own work.

3.1 Ideal Communication-Less Widening

The goal of communication-less Widening is to split the search space so that each worker investigates a different set of models without the necessity of communication. Below we discuss the required properties of such a partitioning, so that communication-less Widening is achieved.

The first property is the *closure property* of such partitions:

Definition Let $M \subseteq \mathcal{M}$ be a set of models. Given a refinement operator r the set of models M is ***closed*** under $\succ_r$ iff $\forall m \in M \, : \, r(m) \subseteq M$.

In many cases, even if the refinement sets generated by the refinement operators have intersections, we are interested that the models chosen by the selection operator at each step are not the same. This may cause duplicate work for the parallel workers, because of potential overlap of the generated refinement sets, but still results in different final solutions for different parallel workers.

Definition Let $M \subseteq \mathcal{M}$ be a set of models. Given a refinement operator r and a selection operator s, the set of models M is ***weakly closed*** under $\succ_r$ and a given data set D iff $\forall m \in M \, : \, s\left(r(m)\right) \in M$.

We include the standard definition of a partition:

Definition Given a family of models $\mathcal{M}$, a division into k subsets $M_1, \dots, M_k$ is called a ***partition***

iff $M_1 \cup \dots \cup M_k = \mathcal{M}$ and $\forall i, j = 1, \dots, k \, : \, i \neq j \Rightarrow M_i \cap M_j = \emptyset$.

Using these definitions, we define Ideal Widening as an partition of the search space in such a way that each parallel worker explores only its assigned partition. Be.

Definition Given a family of models $\mathcal{M}$ and a refinement operator r, the partitioning $M_1, \dots, M_k$ is called a ***closed partition*** iff $\forall k' = 1, \dots, k \, : \, M_{k'}$ is closed under $\succ_r$.

However, such a partitioning is not realistic for many algorithms, where the search starts from a trivial solution, an empty model, so this model needs to be a part of each assigned subset, which means that there will be an intersection between the assigned subsets.

3.1.1 Approximate Partition-Based Widening

What we are interested in is creating such subsets for each parallel worker, so that each model is reachable by at least one parallel worker, using the refinement/selection iteration. Namely, each model needs to be reachable by at least one search path in the search space.

Definition Given a family of models $\mathcal{M}$ and a refinement operator r, a subset $M \subseteq \mathcal{M}$ is called a ***path-closed set*** iff $\forall m' \in M \wedge m'$ is not a base model $: \exists m \in M \, : \, m' \in r(m)$.

Using this, we can define the split of the search space into path-closed subsets which, together, cover the entire search space and each model is reachable in at least one subset:

Definition Given a family of models $\mathcal{M}$, a division into k subsets $M_1, \dots, M_k$ is called a ***path-closed approximate partition*** iff $M_1 \cup \dots \cup M_k = \mathcal{M}$ and $\forall i : \, M_i$ is a path-closed set.

The ideal scenario would be a division into disjoint, closed sets. In practice, a more realistic setup, where the subsets partially overlap but at least each model is still reachable in at least one set can be sufficient. The amount of overlap obviously directly affects potential redundancy when we use this kind of approximate partitioning to widen our search. Figure 3.1 illustrates this approach.

Creating such an explicit, even approximate partitioning will still be difficult for many types of models and algorithms, because it requires prior knowledge and special properties of the search space structure.

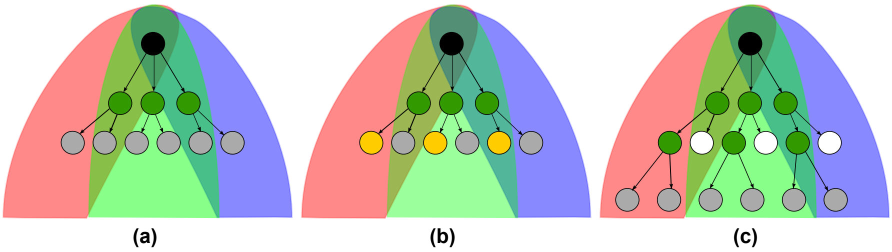


Figure 3.1: Approximate Partition-Based Widening. The search space is divided into a set of overlapping subsets, however, we guarantee that each model is reachable within at least one subset. Instead of selecting from a set of intermediate candidates, each (parallel) worker now only selects the one best candidate in its own subset of the search space.

3.1.2 Path-Based Widening

Instead of relying on explicitly defined partitioning, we can also modify the selection operator of the heuristic to be different for each parallel worker, so that each works with an *individualized* selection operator. They ensure that a particular parallel worker takes only those models into consideration for which it is responsible. Each parallel worker can be assigned individualized preferences for different models in the search space, in an implicit fashion.

Definition Given a family of models $\mathcal{M}$, a refinement operator r, and a selection operator s, a linearly ordered set of models $P_s = \{m_1, \ldots, m_n\}$ is called a ***selection path*** iff $\forall 1 \leq i \leq n-1 : s(r(m_i)) = m_{i+1}$ and m_1 is a base model.

The idea of path-based Widening is now that either the refinement operator or the selection operator or both are individualized in such a way that they prefer a specific subset of models, creating selection-path partitions of the search space.

Definition Given a family of models $\mathcal{M}$, a ***path-based subset*** based on selection operator s is the subset $M_s \subseteq \mathcal{M}$ of models that are selection-path-reachable via s, that is, $M_s = \{m : \exists P_s = \{m_1, \ldots, m\} \wedge m_1 \text{ is a base model}\}$.

Definition Given a family of models $\mathcal{M}$, a set of selection operators $s_1, \ldots, s_k$ is called a ***perfect selection operator set*** iff $M_{s_1} \cup \cdots \cup M_{s_k} = \mathcal{M}$ and $\forall i, j = 1, \ldots, k : i \neq j \Rightarrow M_{s_i} \cap M_{s_j} = \emptyset$.

Finding a set of selection operators that are overlap-free is difficult in the general case.

In Chapter 6 we describe how to define redundant-free refinement operators for certain types of problems. Additionally, in the same chapter knowledge of the global structure of the search space is used to define a set of "good" selection operators.

Furthermore, in Chapters 7 and 8, we demonstrate experimentally, that modifying the selection operators locally leads to exploring diverse paths and improves upon the greedy heuristics on average.

3.2 Diversity-Driven Widening

In the previous section, we discussed approaches to ensure our Widening methods explore different portions of the search space with or without required communication among workers.

Of greater importance is a bigger drawback of all beam search style searches, namely their focus on a narrow portion of the search space. Similar to issues known in genetic algorithms (and in this context, often addressed explicitly, e.g. [72]), these search methods tend to exploit areas around local optima rather than globally exploring the search space and potentially also finding the global optimum. Our partitioning-based approaches described above do not encourage a global exploration either, because the partitioning incorporates no further constraints, such as enforcing *diversity* of the models in separate partitions. The result could be that different partitions actually contain fairly similar models resulting in all of our workers finding similar solutions.

Hence partitioning would benefit from an additional diversity constraint, making sure we are not only following the path (or beam) of (locally) best models but truly explore the overall model space to get closer to discovering the globally best model. Note that this turns model selection into a multi-objective problem as we will now be required to balance the performance of the solutions vs. the additional desired diversity.

We will first describe a randomized approach which aims to give each worker an equal chance of finding the best model. Afterward, we strengthen this by enforcing diversity constraints. Similar to before, we will first sketch how an ideal diversity-driven Widening would operate before discussing communication-less alternatives, which are feasible to implement and are able to ensure diverse solutions by either considering the models themselves or by weighing data elements differently.

3.3 Ideal Diversity-Driven Widening

We first discuss an ideal setup which would lead to a diverse set of final models adhering to a certain diversity criterion. This method resembles $Top-k$ Widening but instead of picking the best k models it picks the subset of k models that are diverse *and* have high performance. Instead of choosing models only according to their quality, additionally we need take diversity into consideration, if we want to avoid the parallel workers converging to one segment of the search space.

A variety of measures can be applied to evaluate both the overall quality of the selected models and their diversity. Given a model evaluation function we can aim to maximize the maximum, the mean, the sum, or other properties of the set of models. For diversity, one can consider different measures to evaluate how diverse the set of models is, such as average pair-wise similarity, the sum over all pair-wise similarities, the minimum or the maximum, depending on our goals. In general:

Definition Let $M \subset \mathcal{M}$ be a subset of models, Ψ and Δ a performance resp. diversity function, and k the width-parameter, then the function $s_{\text{topdiv}-k}$ is called ***best-diverse-k selection operator*** if $\forall M' \subseteq M \wedge |M'| = k \ : \ \Psi(s_{\text{topdiv}-k}(M)) \geq \Psi(M') \vee \Delta(s_{\text{topdiv}-k}(M)) \geq \Delta(M')$,

that is, there exists no other subset of size k that performs better, and, at the same time, is more diverse. For most multi-objective optimization problems, the entire Pareto front contains non-dominated alternatives.

A simple and efficient approach is a threshold-based picking scheme [50], where one iteratively picks the next best model that is at least a threshold away from all previously chosen models. But this approach requires very heavy communication between the parallel workers, to share information and determine the sets of best diverse and high quality solutions, the Pareto front members. The two functions can be combined, turning this into a single optimization problem:

$$\Psi(s(\mathcal{M})) + \alpha\Delta(s(\mathcal{M})) \rightarrow \max . \tag{3.1}$$

Depending on the definitions of the functions Ψ, Δ and parameterα the function can describe Widening with different properties. For example, for $\alpha = 0$, one has the $Top-k$ Widening approach without any diversity implemented. If by Δ is trying to maximize the average pairwise diversity, the behavior of the search would be different than if one is trying to maximize the maximal pairwise diversity (perhaps not so suitable for our goals), and so on.

We will demonstrate in the next chapters types of diversity-driven Widening using differently defined Ψ, Δ and different value of parameter α in the Expression (3.1).

3.3.1 Communication-Less Diversity.

If we do not want to rely on global communication when selecting a high-performance and diverse set of models, we need to ensure that each worker has its own individual strategy for selecting the next model, and that consequently explored selection paths are diverse. We can do this with individual selection operators, which are pre-programmed before the search to have different and diverse search behavior.

This essentially splits the ideal best-diverse-k selection operator into individual operators for each worker. Ideally, the individual selection operators pick a set of models that is at least as good and at least as diverse as if the best-diverse-k selection operator was used. In reality, of course, we can only hope to approximate this idealistic scenario.

For models such as the set covering examples described in [7], we can encourage diversity by making sure that each worker uses a different preference ordering of model fragments. Alternatively, one can also assign different preferences for different data points, to achieve data-based diversity.

3.4 Difficulties of Explicit Partitioning

This is based on [86], and represents the author's own work. Ideal and approximately ideal partitioning are impossible to implement without knowing apriori the structure of the search space. In order to achieve approximate ideal partitioning, each M_i has to be closed under refinement.

3.4.1 Simple Example: Communication-less Diversity via Globally Assigned Preferences (Hashing).

We want to achieve implicitly Widening of width k of the search in a communication-free manner by *individualizing* the k selection operators $s_i, i \in \{1, \ldots, k\}$. This means that each s_i has individualized preference for a different subset of models. The search space of models is unknown in the beginning of the search, so these individual preferences need to be defined implicitly and encoded prior to the search. Due to these restrictions, one cannot use preferences for whole models, but needs to use preferences for model components or data points, since they are what is known apriori and thus define model-based or data-based diversity-driven Widening.

The simplest, most intuitive way to do that, is to assign individualized preferences to different model components or data points directly in a static way. As already stated, the preferences can be based on characteristics of the models, (model-based diversity) or on data points, (data-based diversity). Figure 3.2 illustrates two different selection paths generated by two different selection operators s_1 and s_2.

More formally we can define this simple communication-less Widening as follows.

Given an intermediate model m, ψ_i evaluates the refinement m' based on the original quality measure and an individual preference π_i for m':

$$\psi_i(m') = \psi(m') + t * \pi_i(m').$$

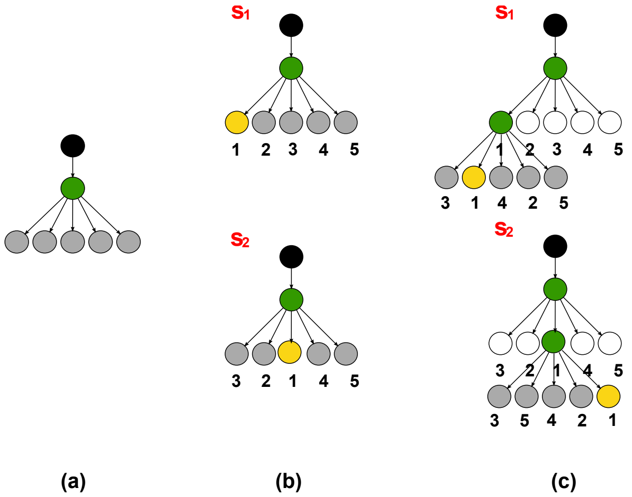


Figure 3.2: Communication-free Widening: two different selection paths generated by two different selection operators s_1 and s_2.

Our goal is to have k different (and diverse) preference orders $\pi_1, \ldots, \pi_k$, so that the selection operators are steered to explore different and diverse paths in the search space. This approach is implemented and explained in detail in Section 7.3.3.

Definition Given a set of parallel workers $\{w_1, \ldots, w_k\}$, communication-less Widening with **diversity based on global preferences** is the set of selectors $s_{hash} = \{s_1^{hash}, \ldots, s_k^{hash}\}$ whose individualized quality measure functions $\{\psi_1, \ldots, \psi_k\}$ assign to models in $\mathcal{M}$ different preferences given by the functions $p_i : \mathcal{M} \rightarrow \mathbb{R}$ to implement a set of diverse partial orders on $\mathcal{M}$ $\{(\leq_i, \mathcal{M})\}$. These preferences for particular models are based either on model properties, or on the data.

The balance between diversity and quality of the explored models can be optimized if the individualized model evaluation functions $\{\psi_1, \ldots, \psi_k\}$ combine the original model evaluation function ψ and the individual preferences for the models. Consider the $\Delta(s_{\text{hash}}(r(m))$ of this Widening approach at a given step. Unlike in diverse $Top-k$, where a threshold guarantees a lower bound for $\Delta(M)$, here one cannot estimate $\Delta(s_{\text{hash}}(r(M)))$. The hope is that by assigning different preferences to parallel workers, they will traverse different selection paths, but the effect is obviously hard to predict and this approach cannot not even guarantee that

$$\forall i \neq j : s_i^{\text{hash}}(r(m)) \neq s_j^{\text{hash}}(r(m)),$$

that is, different workers do not consider the same model. Despite the lack of guarantees, this simple approach leads to surprisingly great improvement in the solution quality. This can be seen Chapters 7 and 8.

Weaknesses of the Simple Communication-less Widening Approach.

This type of Widening leads to an unstructured way to explore the search space. One of the goals of Widening is, that the more parallel resources you have (the bigger the value of parameter k), the larger part of the search space you explore. This method does not guarantee this property. First, the parameter t is data dependent, it determines how much importance is given to Ψ and how much weight is given to Δ. The deviation from the greedy option is unpredictable.

In addition, the static nature of the assignment has a flaw in itself. It lacks flexibility and universal applicability. The workers might not be able to choose their statically preferred models, if they are not up for selection. For example, if w_i strongly prefers model m'_i and w_j strongly prefers model m'_j, yet these two models may not be in consideration, so they have to chose not between these two models, but between m'_k and m'_p and for those they might have the same preference, the difference in their exploratory behavior will not be noticeable at all.

3.5 Conclusions.

In this chapter, we discussed different ways of achieving communication-less Widening, by partitioning the search space among the different parallel workers. We discuss explicit methods, such as Ideal Widening, which are impossible to achieve in reality in the general case, due to lack of information with regard to the models at the beginning of the search. We also discussed implicit partitioning, which is achieved through individualizing the behavior of each individual selection or refinement operator.

We showed a simple example of individualizing the behavior of the selection operator of each worker, based on giving individualized preferences based either on data, *data-based diversity*, or on the models directly, *model-based diversity*. This simple method performs very well experimentally, as we will show in Chapter 7, and Chapter8. However, we want a structured way of exploring the search space and introducing diversity with a more predictable properties, in which we have a better control over the behavior of the parallel workers.

Chapter 4

Widening Using Neighborhoods in the Search Space of Models.

4.1 Motivation

As discussed in Chapter 3, a perfect explicit partitioning of the search space requires a prior knowledge of the hierarchy of models. That is, which model is a descendant of which on the refinement/selection path. Assigning preferences to temporary models (based on model fragments or data) as presented in Section 3.4.1 is a simple intuitive way to approach this task. However, as it was explained, two problems exist. Via a parameter, one cannot control in a meaningful/structured way the diversion from the greedy path. Following this approach, it is not even possible to assign preferences in such a way that they guarantee that two parallel workers i, j would choose different models from the same refinement set $r(m) = M^r$.

Let us consider a different approach to assigning preferences to workers, one that can quantify how much a parallel worker is deviating from a locally optimal choice, based on *the number of models*. In Widening via neighborhoods, the parallel workers use local information, just as in the greedy approach, but consider a larger part of the search space at each step.

Let us consider the following characteristics of a search, represented as the iteration between refinement and selection operations. First, once having reached a model m, each parallel worker has access to its full refinement set $r(m)$. Second, the models of a given refinement set differ from each other by one model fragment. Third, given that the total number of model fragments is n, the subspace of the search space, $\mathcal{M}^l \subset \mathcal{M}$ consisting only of models of size l is the union of all the refinement sets for all models of size $l-1$. Given these observations, we will focus the local behavior of the individual parallel workers in each refinement set, and consequently the model fragments from which step by step the models are built via the refinement operation.

In order to assess the similarity between model fragments and between models, we will consider the topological notion of *neighborhoods* of models, and in particular of the locally optimal model.

4.2 The Metric Space of Model Components

Before we talk about neighborhoods of models, let us discuss the similarity and the dissimilarity of model fragments, the building blocks of a model. During the iterative application of the refinement operator, a solution is built by adding more model fragments to it. Depending on the type of refinement operator $r(\cdot)$ this can be simple model fragments or compound ones.

In the case, when the refinement operator uses only simple model fragments at each step, they are known at the beginning of the search and can be used to encode desired behavior to the individual selection operators prior to the search. That is why, depending on the refinement operator $r(\cdot)$, the Widening problem has different complexity. For different model types, this representation is different. Some, like unordered rules, are simply a combination of its model components, without hierarchy or order playing any role. Others, like decision trees, depend on hierarchy.

Due to the fact that in a given refinement set the models differ by a single component, models from each refinement set can be assigned to different parallel workers prior to the search, without these models being explicitly known, just using the model components. This assignment happens locally, for each refinement set of a given model, without requiring communication between the parallel workers.

We will denote the set of model fragments as X. Given a chosen metric d over the set of model fragments X, we can define a metric space (X, d). Based on the metric d, we can assess the similarity and dissimilarity between two model fragments.

Typically, the metric d will use the data D in order to assess similarity, which constitutes a *data-based* approach to similarity. However, it can also rely on purely model-dependent properties, which constitutes a *model-based* approach. Using the notion of locality and its related concept of neighborhoods, we can divide the space of model fragments into neighborhoods of model fragments, sufficiently similar to each other.

4.2.1 Neighborhoods in the Space of Model Fragments

We can define a neighborhood of model fragments in various ways to serve our different purposes and the properties of the resulting partitioning will be different. Below we present several definitions of this concept, which serve different purposes. Let us start with the most common definition, that is based on a radius from a point.

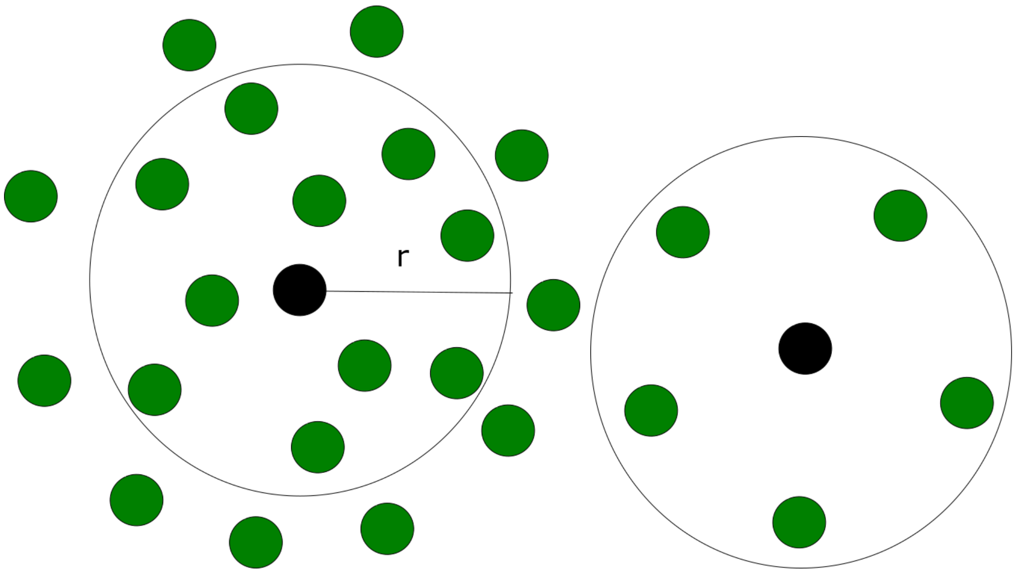


Figure 4.1: Neighborhoods in the metric space of model fragments; r-neighborhood and k-neighborhood $k = 6$.

Definition Let X be a set of model fragments, and let d be a metric. Given the metric space (X, d), and given a radius $r > 0$ the r-neighborhood N_r of a model fragment x is the set of all model fragments in X that are at distance less than r from x.

Model fragments, which are within a neighborhood are *neighbors*, based on the metric d.

For the goals of Widening (and partitioning), when we consider neighborhoods, we are not always interested in neighborhoods of a fixed radius r but may be more interested to use neighborhoods, based on a particular number of model fragments. We will refer to a neighborhood of a model component x of size k as the *k-neighborhood of x*.

Definition Given the metric space (X, d) and an integer $k > 0$, we define a k-neighborhood N_k of a model fragment x, as the ordered set of k model fragments $N_k = (x, x_1, \dots, x_{k-1})$, for which $d(x, x_i) \leq d(x, x_{i+1})$, $i \in \{1, \dots, k-1\} \wedge d(x, x_j) \geq d(x, x_i) \forall x_i \in N_k, x_j \in X \setminus N_k$. Ties are broken randomly.

In other words, the k-neighborhood of a model fragment x is the set consisting of the element x itself and the $k - 1$ model fragments, closest to x according to metric d.

Using the information provided by the neighborhoods, about which model fragments are similar, we can partition the space of model fragments between the parallel workers. For example, partitioning each neighborhood between the parallel workers enforces that within a given neighborhood of model fragments only one model fragment is selected by each parallel worker. Such partitions are *clustering* and *local sensitivity hashing.* Using the definition of k-neighborhood above we can define the set of individualized selection operators for the parallel workers $1, \ldots, k$, $\{s_1^{\mathrm{N}}, \ldots, s_k^{\mathrm{N}}\}$, with which we can define *Widening via k-neighborhoods.* For each model $m \in \mathcal{M}$, each parallel worker prefers one neighbor in $N_k(s(r(m)))$, and its choice is unique for this neighborhood. Note, that the same model fragment will be a neighbor in different neighborhoods, the neighborhoods are relative to a particular model fragment.

4.3 Neighborhoods of Models in the Subspaces of Refinement Sets.

The concept of locality using neighborhoods over the space of model components translates into the space of models. Notice that defining neighborhoods over the space of model components is equivalent to defining neighborhoods over the set of those models, that are direct refinements of a given model because they differ from each other by one model component.

In a similar fashion, we can use a metric to define neighborhoods in the refinement sets of each model. Given that the refinement operator creates a model only by using simple model fragments at each step, the neighborhoods in the space of model fragments translate to neighborhoods in the subspace of refinements of a given model, M^r. We want to assign to the parallel workers different models, using neighborhoods of models within the metric subspace of model refinements (M^r, d).

Given m and a refinement operator r, we can define a k-neighborhood N_k of the greedy choice $m' = s(r(m))$ as follows:

Definition Given a model m, a selection operator s, a refinement operator r, and a distance measure d, the **k-neighborhood** of $m' = s(r(m))$ is the ordered set $N_k(m') = (m', m'_1, \ldots, m'_{k-1}) \subseteq r(m)$ where $\forall i \in \{1, \ldots, k-2\} : d(m'_i, m') \leq d(m'_{i+1}, m')$ and $\nexists m'' \in r(m) \setminus N_k(m') : d(m', m'') \leq d(m', m'_{k-1})$. Ties are broken randomly.

Or, in simple words, the k-neighborhood of a model m consists of $k - 1$ refinement neighbors that are nearest according to the distance measure d.

Using the definition of k-neighborhood above we can define the set of individualized selection operators for the parallel workers $w_1, \ldots, w_k$, $\{s_1^{\mathrm{N}}, \ldots, s_k^{\mathrm{N}}\}$, with which we can realize a *Widening via k-neighborhoods.* For each model $m \in \mathcal{M}$, each parallel worker prefers exactly one neighbor in $N_k(s(r(m)))$, and its choice is *unique* for this neighborhood. For visual representation see Figure 4.2.

Since we are interested in Widening of a greedy heuristic, which is focused on selecting always a locally optimal choice, we define the neighborhoods with respect to the

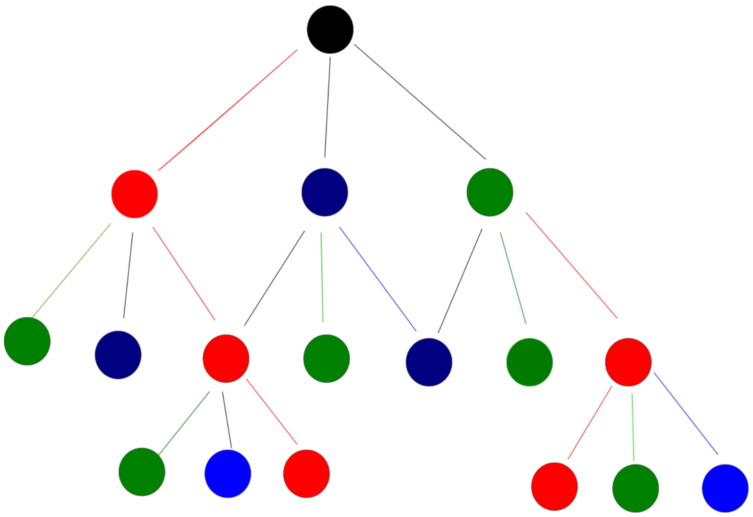


Figure 4.2: Schematic representation for Widening via local partitioning of $M^r = r(m)$ using k-neighborhoods. Different neighbors are assigned to different parallel workers, represented by their color. Here $k = 4$. Each parallel worker chooses the model assigned to it apriori.

greedy choice. This, however, is not necessary – we can partition of a refinement set M^r irrespective of the locally optimal choice.

Definition Given a family of models $\mathcal{M}$ with a refinement operator r, we define *randomized k-neighborhood Widening*, Nr_k^o, as optimality k-neighborhood Widening, where each member of a given k-neighborhood is selected by a parallel worker with equal probability $\frac{1}{k}$.

Randomized Widening using Neighborhoods

The most trivial way to apply neighborhood-based Widening in this scenario is to randomly assign the models from each M^r to parallel workers, without taking into consideration any properties of the models from M^r and at each step the parallel worker chooses a random model from its partition from the given M^r. This will result in the randomized exploration of the search space.

4.3.1 Neighborhood Size

The size of the neighborhood controls the amount of straying from the locally optimal, greedy solution. Depending on the goal, different sizes are advantageous. Heavy intersections between neighborhoods are more likely for Widening with k-neighborhoods with small neighborhood size. Then the intersections between the neighborhoods are most likely representative of getting stuck at a local optimum. When using small neighborhoods there is a higher risk to converge to a local optimum, and this risk is increased due to the lack of communication, and thus the inability to remove at least the repeated models, thus increasing the risk of overlapping exploration between the parallel workers.

The larger the neighborhoods, the more the potential intersections approximate the structure of the search space and are not a consequence of a local optimum. For many problems, a model can be reached via different selection paths. This is related to the structure of the model space for a given problem. For example, with the greedy heuristic for the SCP, one can start the search with various initial refinements and still reach the same solution.

On the other hand, increasing the size of the k-neighborhoods over a certain value can lead to a sparse exploration of the search space and strong deviation from the greedy search path. Furthermore, in the general case, this method does not guarantee the reachability of every model. It leads to a sparse exploration of the search space, where many models are not reachable. In the extreme case, in which k is very large, this will give strong preference for Δ compared to Ψ and will start to approximate a randomized search leading to investing parallel resources in the exploration of degenerate solutions. The size of the neighborhood also depends on whether the goal is exploration or exploitation. Assuming ample available parallel resources, in certain situations, it may be beneficial to favor reachability of models within the neighborhoods we are exploring, over investing parallel resources into larger and larger neighborhoods. The k-neighborhoods can be generalized into θ, k-neighborhoods, where θ is the size of the neighborhood, with which we are Widening, and k is the number of parallel resources.

4.4 θ, k-Neighborhoods

Many different models are reachable via selection paths that share common initial subpaths, but then diverge, as shown in Figure 4.3. We want to define another type of Widening via neighborhoods, that guarantees reachability for every model at a fixed level l. In order to achieve that, multiple workers' paths may have to intersect.

Definition Let θ be the size of the neighborhood, and let k be the Widening parameter. Given a model m, a selection operator s, a refinement operator r, and d, a chosen distance measure, a θ, k-neighborhood of $m' = s(r(m))$, $N_{\theta,k}(m')$, is an element of the Cartesian product $N_\theta(m')^k = N_\theta(m') \times \ldots \times N_\theta(m')(k \text{ times})$.

Namely, k models are selected from $N_\theta(s(r(m)))$. If $k \gg \theta$ this implies repetitions between models $m'_0, m'_2, \ldots, m'_{k-1}$.

Widening via θ, k-neighborhoods is presented in Figure 4.3.

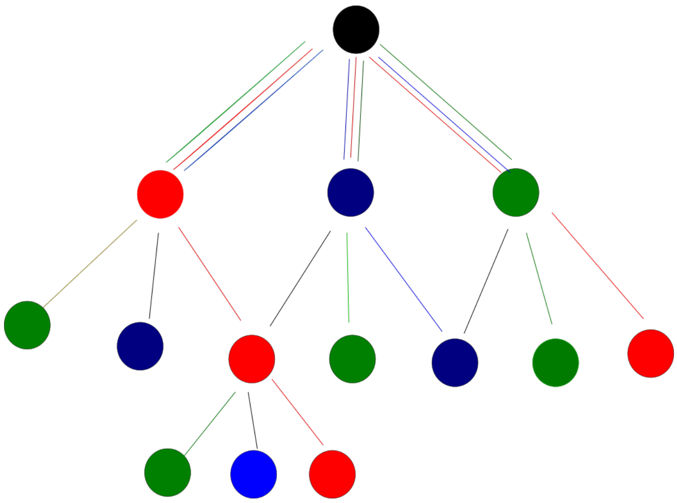


Figure 4.3: θ, k-neighborhood Widening.

4.5 Optimality Neighborhoods

Widening via optimality neighborhoods is an attempt to emulate the $Top - k$ Widening in a communication-less manner. Its goal is to decrease the "greediness" of a heuristic by considering the first, second and so on choices, instead of only the greedy choice. In this type of neighborhoods, the metric is defined as a distance from the quality score ψ of the locally optimal model. We will discuss the properties of the Widening via optimality neighborhoods in Chapter 5. The size of the neighborhood serves as a constraint how much drift away from the locally optimal solution is allowed. Widening via optimality neighborhoods is similar to a randomized beam search, limited to picking k models at random from the top k^l candidates (branches). For very large k, N_k may stray too much away from the locally optimal solutions in a randomized fashion, to be useful. For very small k, just like the $Top - k$ search, it can converge to a local optimum.

4.6 Similarity Neighborhoods

Similarity neighborhoods are k-neighborhoods where the metric d is based on a similarity evaluation of particular properties of the models. Widening via similarity neighborhoods explores solutions with properties similar to those of the greedy choice. Similarity neighborhoods can be used in many different scenarios. For problems where it is known that the greedy algorithm leads to a good solution, exploring the area around the solution of the greedy algorithm can help to discover the optimal one or solutions of even higher quality. At the beginning of the search, a good strategy is to use diversity and explore more of the search space. However, once good areas of the search space are discovered, it is useful to explore these good areas in more detail in order to discover solutions of higher quality (or even the optimal solution). This intensifying of the search in promising areas is referred to *exploitation*. An additional application is the so-called *similarity search*. Many similarity searching strategies already rely on neighborhood-based greedy-like approaches. In certain situations, one may need to discover many similar models with certain properties, which perform well. Incorporating Widening via the similarity neighborhoods to these strategies can further improve the results of these searches. For more details about the properties of Widening via similarity neighborhoods, refer to Chapter 5.

4.7 Diversity for Neighborhoods

In this section, we will discuss only the application of diversity to Widening via optimality neighborhoods. Widening via similarity neighborhoods has a different goal: the search of a set of models that share similar properties, based on a given similarity measure. Widening via optimality neighborhoods, in contrast, aims at exploring different parts of the search space and avoiding local extrema. Widening using (optimality) neighborhoods alone does not guarantee diversity of exploration, due to the fact that similarity neighbors can be also optimality neighbors in a given neighborhood. By building *diverse* neighborhoods, consisting of optimal, yet diverse solutions, the parallel workers will be forced to explore a larger part of the search space. We will show in Chapter 5 that applying diversity locally leads to a globally more diverse exploration, without the need for communication. This is experimentally demonstrated in Chapters 7 and 8.

Increasing the size of the neighborhoods and thus increasing the randomization of the search, will not add diversity in the best possible way. The background distribution of the search space may be such, that randomized search does not lead to exploring diverse solutions. This, for example, can happen if the background distribution is strongly non-uniform. This is why using diversity explicitly is preferred. Furthermore, taking into consideration the performance of the models is also important. Using only diversity can lead to exploring *different*, but not necessarily *promising* areas of the search space.

4.7.1 Diversity Methods Inspired by Niche Genetic Algorithms: Fitness Sharing and Crowding

In the field of genetic algorithms (GAs) converging too early to a local optimum, typically because of insufficient genetic variation, is very well investigated problem. A class of approaches, called *niching*, are developed with the goal of maintaining diversity within a population. This problem is very close to the problem one potentially encounters when widening a heuristic, without ensuring diversity. Without diversity, the widened algorithm may explore similar, or equivalent search paths, which is similar to losing diversity in the population in the scenario of genetic algorithms, and as a consequence, converge to a suboptimal solution. The methods used for niching in the case of GAs can be naturally implemented in our case. This section is based on [111] and describes how different niche approaches can be applied in the context of Widening.

Fitness Sharing

Fitness sharing[111] is one of the most prevalent niching approaches. It aims to promote variety within a population with high fitness throughout the search, by forcing similar individuals to share their fitness. The individual's fitness is negatively proportional to the number of individuals in its niche, so the more common is an individual in a population, the more its original fitness is degraded. The more common an element is (the more similar to others), the more its original fitness is penalized.

In our context, we can look at temporary models generated at a given step as new individuals in a population. We try to increase or preserve the variety in the population. Fitness sharing can be applied to model fragments as well as to models (refinements). Given a set of model refinements $M = \{m_1, \ldots, m_n\}$, a model quality measure ψ and a metric d, a modified model quality measure, based on ψ, which incorporates fitness sharing, ψ_{fsh} is defined as

$$\psi_{fsh}(m_i) = \frac{\psi}{\sum_{j=1}^{n} sh(d(m_i, m_j))},$$

$$sh(d) = \begin{cases} 1 - (\frac{d}{\sigma})^{\alpha} & \text{if } d < \sigma \\ 0 & \text{otherwise.} \end{cases}$$

In this scenario, the parameter σ is a distance threshold, below which the models are considered to be in the same niche. The parameter α controls how much is the original model quality influenced by the niche count of a given model. This method indirectly forces a neighborhood of an optimal refinement m' to consist of more diverse neighbors, compared to the simple optimality neighborhood approach, because the quality of a given model is evaluated based also on the "rarity" of the model, and not just its quality. However, this approach allows no direct control over the diversity of selection, compared to direct "individualized" preference modification of each selection operator. Another negative aspect of this method is that it is parameter-dependent, and good values for parameters σ and α can be determined only experimentally. Additionally, this approach has high computational costs. The benefits of this approach are that it is data-intelligent and provides a full exploration of the refinement sets and peak selections by building the niches.

Crowding Methods

Crowding methods maintain diversity by replacing similar elements with a new one. Various crowding-based approaches exist. We will consider the following crowding methods and their application for diversifying optimality neighborhoods. In *standard crowding* [39] only a percentage, called generation gap (G), of the total population, has offspring and dies in each generation. The new offspring is produced by the operations of mutation and mating. A subpopulation of size CF(crowding factor) is picked at random from the total population and the individual most similar to the offspring is replaced by the offspring. Offspring are generated by mutation and mating. The individual, most similar to the offspring within a randomly drawn subpopulation is replaced by the offspring. In *deterministic crowding* [110], further improvement is introduced by a fitness-based competition between the parents and offspring and the parent is substituted only if its' quality is lower.

How can we use the crowding approaches in generating diverse neighborhoods of models? The parent-offspring tournament is not applicable in the refinement/selection setting, because a refinement is at least as good as the original model. The tournaments that make sense in this context are competitions between the models, which are similar, that is, from the same subpopulation (niche).

4.7.2 Diversity Using Simple Threshold. Diversity via Similarity Neighborhoods.

In multi-objective optimization [144], there is rarely a solution, which minimizes all functions at the same time. Therefore, one searches for Pareto optimal solutions, namely, such solutions that cannot be improved in any objective without at the same time getting a worse value for at least one other objective.

Definition For a multi-objective optimization problem $\max(f_1(M), f_2(M))$, solution $M^{'}$ is said to dominate a solution $M^{''}$ if $f_i(M^{'}) > f_i(M^{''})$ and $f_j(M^{'}) \geq f_j(M^{''})$, where $(i, j) = (1, 2)$ or $(i, j) = (2, 1)$. Adapted from [144].

In our settings, the multi-objective optimization deals with the diversity and optimality of the solution set, $\max(\Delta, \Psi)$.

Definition A solution $M^* \in \mathcal{M}$ is called Pareto optimal, there does not exist any solution, which dominates it. A Pareto front for a given a multi-objective optimization problem comprises the full set of Pareto optimal solutions. Adapted from [144].

In order to build a diverse optimality neighborhood of the optimal model in a given refinement set M^r, one can use a simple diversity threshold δ, based on a similarity measure. The corresponding set of solutions will be the set with the higher Ψ from all possible sets of models, which fulfill the diversity threshold δ. This set of solutions will be a non-dominated solution from the Pareto front for Problem 3.3 in Chapter 3. However, this set is non-dominated only locally, within the given refinement set. In contrast, the methods, which use communication, such as diverse $Top - k$, have access to all k refinement sets of the models, chosen on the previous step.

Instead of using a fixed threshold, which is data dependent, we can take another approach for building such a locally non-dominated set. For a structured, data independent strategy we can use the already defined similarity neighborhoods. We simply want to use the combined information from similarity and optimality neighborhoods in order to achieve our goal. One way to do this is to choose the best performing models which do not belong to the same similarity k-neighborhood. In such a way, a diverse neighborhood of models, which are both, of high performance, yet dissimilar in terms of properties, is formed.

4.7.3 Diverse k-Neighborhoods versus Large k-Neighborhoods

Increasing the size of the neighborhood, relative to the number of parallel workers, is the most trivial way to add diversity to the exploration. We will discuss this in more detail in the next chapter. We will see that simply increasing the size of the neighborhood, while runtime efficient, does not lead to exploring diverse-and-promising solutions, but leads to randomized exploration with results, worse than the one obtained by the greedy algorithm. Diverse exploration aims at investigating all different peaks in the search space, which are promising solutions. This is achieved by building diverse neighborhoods while taking model performance into consideration.

4.8 Widening using Neighborhoods versus Ideal Widening

4.8.1 Neighborhoods and Reachability.

By definition, Widening via k-neighborhoods does not guarantee reachability since each neighbor is assigned to only one parallel worker, but it can have multiple refinements. Of these refinements, only the one assigned to the parallel worker in question will be explored. As Figure 4.1 shows, along the yellow refinement path only the yellow refinement is explored, the others are not. In contrast, θ, k-neighborhoods guarantee reachability, given sufficiently large k.

4.8.2 Widening via Neighborhoods and Partitioning of the Search Space

In the general case, partitioning of the search space is not achieved via Widening via neighborhoods. However, choosing a behavior for the parallel workers locally (such as to select good and diverse solutions), leads to obtaining results globally (reach many promising peaks in the search space landscape). Using diversity neighborhoods we can obtain diverse exploration without communication. More will be discussed in Chapter 5, which focuses on the properties of Widening via neighborhoods. The local methods perform in practice, but in order to have guarantees for the properties of the solutions, we need *global* approaches to diversity, as discussed in Chapter 6.

We will show in Chapter 5 that path-based partitioning can be achieved through Widening via θ, k-neighborhoods and that it allows for reachability for each model. This approach, however, leads to redundancy in the obtained solutions as well as exploring multiple nonpromising solutions (which is the majority of the search space).

4.8.3 Widening via Neighborhoods and Diversity-driven Widening

Ideal diversity Widening was described in Chapter 3 as one where the parallel workers pick a solution, which is a set that is non-dominated according to optimality and diversity. Namely, a solution, which is part of the Pareto front for the multi-objective optimization problem 3.3. Consider, diverse $Top-k$ Widening with communication between parallel workers. At each step from all the explored refinement sets $M = \cup r(m_i)$ the method is allowed to choose a set of models from the superset M. In contrast, Widening via diverse neighborhoods approach is based only on selecting a non-dominated solution set from the Pareto front for Problem 3.3 in each refinement set. We will show in the next chapter that introducing diversity locally in neighborhoods leads to increasing diversity globally in the exploration, without the need for communication and how locally Pareto non-dominated solution sets perform globally.

4.9 Mixed Strategies

As discussed, different neighborhood approaches have different applications due to their different properties. Widening via optimality neighborhoods, especially diverse neighborhoods, is focused on exploration, looking for good solutions in different parts of the enormous search space. Widening via similarity neighborhoods aims at intensified exploitation of a promising subspace of the search space. Both exploration and exploitation can be combined. Namely, one can use an exploratory approach at the beginning of the search, and upon discovering good solutions, one can use exploitation by employing Widening via similarity neighborhoods.

4.10 Complexity Considerations

A big issue of communication-less Widening via neighborhoods is the running time of the methods and how it compares to the running time of the greedy heuristic. While not having the communication overhead problems, in many approaches additional necessary computations and/or preprocessing can affect the running time of the algorithm (in some cases, greatly). The running time of the greedy heuristic depends on three main operations: creating the refinements, evaluating the created refinements, and selecting the best refinement. The runtime complexity depends on the complexity of these three operations, performed at each step, and on the number of steps l necessary to find the solution. Widening via neighborhoods requires that each parallel worker builds its neighborhood (of size θ). Depending on the type of the neighborhood, the complexity of building such a neighborhood can be different, but for certain greater than that of the greedy algorithm.

4.10.1 Widening via Optimality Neighborhoods

For optimality neighborhoods, we need to do a sorting operation at each step, and we need to find not just the best, but the θ best solutions. Data structures such as a priority queue can help make the sorting more efficient. Moreover, Widening via optimality θ-neighborhoods requires preprocessing, which assigns θ labels to each model fragment, which takes $O(n)$ time. In the case of optimality θ-neighborhoods, the selection step includes looking up the label for a given neighbor, which can be done in constant time.

Thus, Widening via optimality neighborhoods has a running time very close to that of the greedy algorithm.

4.10.2 Widening via Similarity Neighborhoods

For Widening via similarity neighborhoods, the running time depends on the implementation chosen. Without special preprocessing, each parallel worker needs to "build" the neighborhood of the locally optimal model at a given step on the go. Building a similarity neighborhood requires $n - l - 1$ comparisons – at each step l the optimal model refinement needs to be compared with all other possible $n - l$ refinements and then θ most similar model refinements are chosen. The running time depends greatly on the cost of calculating of the similarity measure used, and on the dimensions of the data, as well as on the number of refinements of a model at step l. In this approach, the size of the neighborhood does not influence the running time, because at each step all the refinements are compared to the optimal neighbor, and the θ most similar are chosen, but the same number of comparisons are performed.

4.10.3 Widening via Diversity Neighborhoods

Adding diversity to optimality k-neighborhoods can potentially have a great impact on the runtime, depending on the type of method used. Diversity neighborhoods built using a simple threshold require that each new refinement is compared to all the previously selected refinements. Thus, the threshold-based methods require the algorithm to perform at least $\frac{\theta(\theta-1)}{2}$ comparisons at each step.

Moreover, the running time depends on the number of comparisons needed to be performed, as well as the cost of evaluating each comparison. The cost of each evaluation depends on the type of distance measure used, how expensive it is to calculate it, as well as on the dimensions of the data. The number of comparisons depends on the size of the neighborhood θ, and on the size of the diversity threshold δ, as well as on the total number of possible refinements at level refinement l. A higher threshold will lead to more comparisons (due to more models failing to satisfy the threshold requirement). A larger size of the neighborhood will affect as well the number of comparisons needed. In the worst case, for a very difficult threshold δ, $\theta(n-l)$ comparisons need to be made at each step, k times in parallel. As the neighborhood size increases, $\frac{(n-l)(n-l-1)}{2}$ ($O(n^2)$) comparisons may need to be made at each step.

4.10.4 Number of Iteration Steps (l)

The runtime also depends on the number of necessary steps for each parallel worker to find a good-enough solution. During Widening, some parallel workers may find sufficiently good solutions faster than the greedy algorithm, but some may do so slowlier. The slowest parallel workers in Widening will do worse than greedy. This includes especially the diverse neighborhoods, where due to diversity many of the explored paths will not be sufficiently good and may take a long time until they discover a satisfactory model.

4.10.5 Preprocessing and Running Time

The preprocessing for similarity neighborhoods requires building for each model refinement ($|X| = n$) a table with its θ most similar model refinements. This will take $O(n^2)$ comparisons to build. A $O(nd + kn)$, where d is the dimension of the space of models. After the preprocessing, the running time of Widening via similarity neighborhoods is equivalent to Widening via optimality neighborhoods, the lookup for the k similarity neighbors can be done in constant time.

Preprocessing using $k-d$ trees or local sensitivity hashing and other methods, which assign for each model fragment its neighbors at the beginning of the search, can be helpful in improving the running time for both similarity and diversity-based neighborhoods. Depending on the properties of the given problem, a different approach may be beneficial.

4.11 Conclusions

Widening via neighborhoods is a local communication-less approach to diversity-driven Widening. It refers to assigning differed and/or diverse models from each refinement set $M^r = r(m)$ of a model m. The goal is to use this local behavior over each refinement set in the search space in order to achieve the desired global behavior of the parallel workers. Depending on the type of metric used, as well as on the relative magnitude of the number of parallel workers and size of the neighborhood, these Widening approaches have different properties. In Chapter 5 we investigate whether by locally enforcing a behavior on the refinement sets, a globally diverse Widening of the search can be achieved, without the need for the parallel workers to communicate with each other. Different types of neighborhoods of the locally optimal model can be used. Furthermore, for each model, the refinement set of that model can be partitioned without the consideration of the locally optimal model.

Chapter 5

Local Approaches: Properties of Widening via Neighborhoods for Refinement Operators of type 1

Parts of this chapter are accepted for publication as [89].

5.1 Search Space Graph $G_{\mathcal{M}}$

In this chapter we will refer to a direct refinement simply as a refinement. Let $\mathcal{M}$ be a family of models, X be the set of model fragments in $\mathcal{M}$, r be a refinement operator over $\mathcal{M}$. We can use $\mathcal{M}$ and the refinement operation $r(\cdot)$ to define a graph $G_{\mathcal{M}}(V, E)$, where V is the set of vertices, and E is the set of edges, defined as follows: $v \in V \Leftrightarrow v \in \mathcal{M}$ and $\forall m, m' \in \mathcal{M}, m' \in r(m) \exists e(m, m') \in E$.

A general refinement graph is shown in Figure 5.1. Figure 5.2 demonstrates a Widening via neighborhoods in the graph $G_{\mathcal{M}}$. The properties of the Widening methods will depend on the structure of the search space.

Clearly, $G_{\mathcal{M}}$ is a DAG.

Each refinement operator, which at each refinement step for each temporary model generates all possible refinements by adding only *simple* model fragments, will be associated with a refinement graph $G(V, E)$, in which each node at refinement level l can be reached by an equal number of paths. This will hold in the simplest case for models such as decision rules, item sets, where there models are unordered sets of model fragments, as well as for decision tree algorithms, where there is hierarchy between the model components, which build a given model.

Lemma 5.1.1 *A refinement operator, which at each step for each temporary model generates all possible refinements by adding only* simple *model fragments, will be associated*

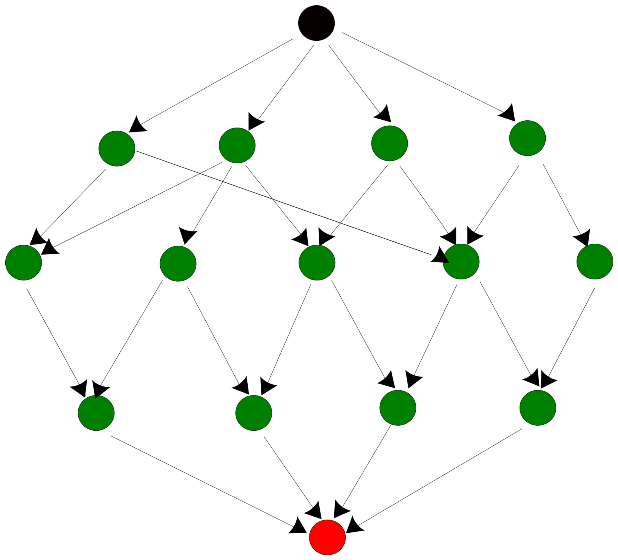


Figure 5.1: A general refinement graph $G_{\mathcal{M}}$ for $\mathcal{M}$.

with a refinement graph $G(V, E)$, in which each node at refinement level l can be reached by an equal number of paths. Holds for algorithms like decision trees, decision rules, association rules, the set cover algorithm, and others.

Proof Each model at level l can be reached by an equal number of paths, because each model at level l can be built in the same number of ways. This follows from the fact that the refinement operation uses only simple model fragments.

5.2 The Search Space Graph as a Lattice for Family of Models $\mathcal{M}$ with a Refinement Operator r of Type 1

Different types of refinement operators exist, depending on their complexity. The type of refinement operator defines a particular structure of the search space.

Definition Let $\mathcal{M}$ be a family of models, X be the set of model fragments in $\mathcal{M}$, r be a refinement operator over $\mathcal{M}$ with the following two properties: only one model fragment is added at a single refinement operation, and the order, in which the model fragments are added, does not matter. Namely, the set of model fragments $\{x_1, \ldots, x_l\}$ uniquely defines a model m and $\forall m' \in r(m) : m' \setminus m = x', x' \in X$. We will refer to such a refinement operator r as *refinement operator of type* 1.

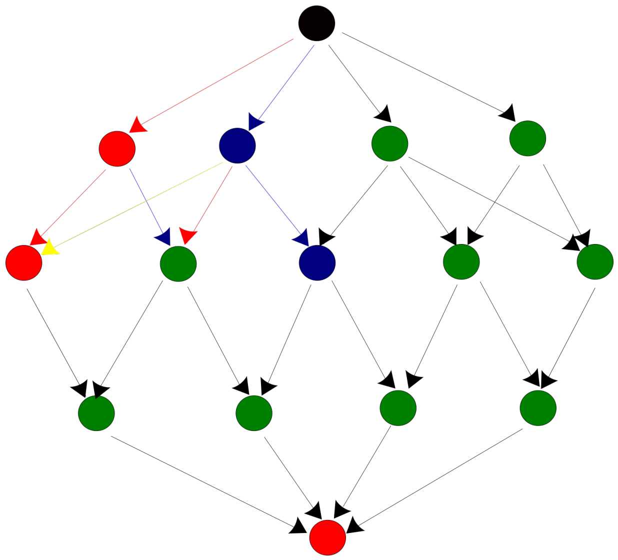


Figure 5.2: A representation of N_k^o for $k = 2$ in the search space graph of $\mathcal{M}$.

We will discuss the search space properties for this most basic type of refinement operators.

First, let us give a general definition of the mathematical structure lattice.

Definition Let S be a set. We will call S a lattice, if there is a partial order on S, in which each two elements have a unique supremum element and a unique infimum element.

A simple lattice, a graph for a family of models for refinement operator of type 1 and 3 model refinements, is presented in Figure 5.3.

Lemma 5.2.1 *Let $\mathcal{M}$ be a family of models, with refinement operator r of type 1. Then, $\mathcal{M}, \leq$ defines a lattice, where $\leq$ is the partial order defined by r on $\mathcal{M}$.*

Proof Let X be the set of model fragments on $\mathcal{M}$. Then $\mathcal{M}$, given that r is of type 1, is the powerset 2^X. It is a known fact that the power set of a set forms a lattice, and we will show it below. First, we will show that each two nodes have a unique supremum. Consider two models $m_i = \{x'_1, \dots, x'_k\}$ and $m_j = \{x''_1, \dots, x''_l\}$. Then their supremum is $sup(m_i, m_j) = m_i \cap m_j$. Their unique infimum is $inf(m_i, m_j) = m_i \cup m_j$.

Lemma 5.2.2 *Each node (model) at refinement level l will be of size l and will have l in-degrees.*

Proof This follows from the definition of the refinement operator of type 1.

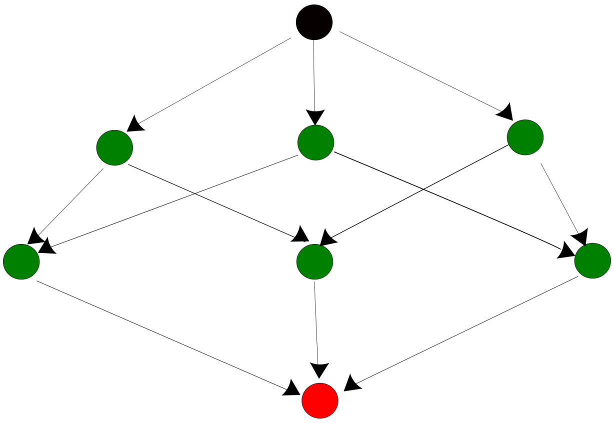


Figure 5.3: A search space graph, lattice, for $\mathcal{M}$, where the refinement operator is of type 1 and the number of model fragments is 3.

Lemma 5.2.3 *The lattice of the family of models $\mathcal{M}$, $L_{\mathcal{M}}$ is a distributive lattice.*

Proof A lattice of sets, where the lattice operations can be given by set union and intersection, is always distributive due to the properties of these operations.

In fact, for refinement of type 1, the lattice is known as the power set lattice. It is known that power set lattices are Boolean lattices, [143]. The opposite does not always hold.

5.2.1 Example: Search Space Graph for the Set Cover Problem (SCP)

Formal Definition of the SCP

A detailed description of the set cover problem and the greedy algorithm and its Widening is presented in Chapter 7. Here we will discuss it only briefly in the context of the refinement graph $G_{\mathcal{M}}$ of the search space. We consider the standard (unweighted) set cover problem. Given a universe X of n items and a collection $\mathcal{S}$ of q subsets of $X : \mathcal{S} = \{S_1, S_2, \ldots, S_q\}$. We assume that the union of all of the sets in $\mathcal{S}$ is X, with $|X| = n$: $\bigcup_{S_i \in \mathcal{S}} S_i = X$. The aim is to find a sub-collection of sets in $\mathcal{S}$, of minimum size, that covers all elements of X. A model m in this setting is a collection of subsets, or a cover C. The refinement operator $r(\cdot)$ adds a single subset, not yet part of C, to C.

The Lattice $G_{\mathcal{M}}$ for SCP

At level $l = 0$ we have only the empty model $G^0 = \{m_0\}$, $m_0 = \{\}$. At level $l = 1$ the graph consists of each possible subset, provided by the problem. $G^1 = \{s_1, \ldots, s_j\}, j = q$. The refinement operator $r(m)$ generates all possible refinements, which consist of adding a single subset to m, which does not yet belong to m. At level l the graph G^l will consists of the models of complexity l (i.e. models containing l subsets). The paths between nodes show the refinement relationship.

This search space and its graph structure is useful for our approaches due to its straightforward simplicity and good properties. First, each model at level l can be reached via $P(l, l) = l!$ paths. Second, we know exactly how many models there will be at each level and how many models will contain a single item. The search space is the power set lattice for $n = |X|$.

5.2.2 Example: Lattice of Single Rules

The graph for these search spaces is similar to the one described above. The model fragments X in this case are the *attribute tests* (possible values for the particular attributes, based on the data). The refinement operation in this example consists of adding a single attribute test at a given step. The graph of the hypothesis space, at each level l will consist of hypotheses with l attribute tests, at level $l = 0$, $G^0 = \{m_0\}$. The number of paths to each model in this graph will be $P(l, l) = l!$. The search space is a power set lattice $n = |X|$.

5.2.3 Example: Lattice of Itemsets

In frequent item set mining, a dataset D containing a set of transactions is given. Each transaction has a number of items, the set of all items is I. Frequent itemset mining algorithm finds all common sets of items, based on their support. The support of an itemet i is the number of transactions, which contain i.

Lattice of Itemsets

In this setting, a temporary model is a collection of items. The goal of the algorithm is to find all frequent itemsets, which have sufficient support (above a particular, user-selected threshold). At each step, the refinement operator generates all possible refinements, where a single refinement is generated by adding a single transaction to a collection of itemsets. The full search space graph for this problem is also a lattice. At level $l = 0$, in the lattice is the empty itemset. The level $l = 1$ consists of all the itemsets of size 1, namely, all the possible elements of I. The next level, $l = 2$, of the refinement graph consists of all itemsets of size 2, and so on. In computer science, it is a well known fact, that the structure of the search space for this problem is a lattice, which is actually the power set lattice for $n = |I|$.

5.3 Disadvantage of Widening via Neighborhoods

The drawback of Widening via neighborhoods originate from the method being without communication and local. The lack of communication between the parallel workers deprives them from the collaboration and synchronization which comes from sharing information. The local methods define the behavior of the parallel workers in each set of model refinements $M^r = r(m)$ individually (be it diversity, optimality or similarity). One does not know how different neighborhoods in different refinement sets relate to each other. The parallel workers may reach the same solution via different paths. As discussed in Chapter 4, the goal is to use local behavior, which is based on the refinement set M_i^r of each model m_i, to achieve the desired global behavior of the parallel workers.

5.4 Performance of N_k^o

N_k^o Widening is a communication-less Widening strategy, which aims to explore the search space by considering not just the locally optimal choice but to use also the k optimality neighbors of the locally optimal choice in each refinement set $M^r = r(m)$. It aims to emulate in a communication-less way the $Top - k$ Widening approach. In contrast, $Top - k$ Widening is a communication-heavy approach, which at a given refinement step selects the best k models from $\bigcup M_i^r = \bigcup r(m_i), i = 1, \ldots, k$, where $\{m_1, \ldots, m_k\}$ are the models chosen at the previous step. Each parallel worker in $Top - k$ has access to each of the k refinement sets at a given step, while each parallel worker in N_k^o has access only to one refinement set at a given step.

It is important to see how these two methods compare to each other and whether the communication-less Widening strategy can compete with the communication-heavy $Top - k$.

Using optimality neighborhoods to achieve Widening follows the general approach of the $Top - k$ algorithm, to loosen the greedy property and to look not only for the locally best, but also to consider the best k solutions without the necessity of using communication between parallel workers. For $k = 1$, both methods explore the greedy path, and will obtain the same results. We will study how the two approaches differ for a larger k.

First, let us compare $Top - k$ and N_k^o in terms of their search space exploration.

Lemma 5.4.1 *Let $m_i, m_j \in \mathcal{M}$ be two distinct models, then the optimality neighbor-*

hoods of these two models can have at most one model in common:

$$|N_k^o(m_i) \cap N_k^o(m_j)| \leq 1.$$

In fact, they intersect iff *the two models belong to the same refinement set* $m_i, m_j \in r(m)$.

Proof The statement follows from the lattice property. Every two nodes have exactly one supremum and one infimum. The infimum can be a direct refinement of both models or a refinement, reached by several applications for the refinement operator.

Let us consider the artificially constructed Widening approach $FullTop - k$.

Definition Given a model evaluation function $\psi : \mathcal{M} \to \mathbb{R}$, and models $m_1, \ldots, m_k$ the function $s_{\text{FullTop}-k}$ is defined as follows:

$$s_{\text{FullTop}-k}(r(m_1, \ldots, m_k)) := \bigcup_{i=1}^{k} s_{\text{Top}-k}(r(m_i))$$

.

$FullTop - k$ search is essentially a breadth first search with pruning to the first k children of each already explored node (model). We will use $FullTop - k$ to bound the subspaces of the search space explored by both $Top - k$ and N_k^o and compare them.

Lemma 5.4.2 *The following two conditions hold.*

1. $Top - k(\mathcal{M}) \in FullTop - k(\mathcal{M})$.
2. $N_k^o(\mathcal{M}) \in FullTop - k(\mathcal{M})$.

Proof Part one follows by design. More precisely, $Top - k$ selects the best k models from $\cup r(m_i)$, $i \in \{1, \ldots, k\}$. In the extreme, these are k models from the same refinement set $M^i = r(m_i)$.

Part two also follows from the design: N_k^o explores exactly a subset of the paths, traversed by $FullTop - k$.

The relationship between $Top - k$, N_k^o, and $FullTop - k$ is visualized in Figure 5.5.

Definition Given a family of models $\mathcal{M}$ with a refinement operator r of type 1, we define *randomized k-neighborhood Widening*, Nr_k^o, as optimality k-neighborhood Widening, where each member of a given k-neighborhood is selected by a parallel worker with equal probability $\frac{1}{k}$.

Instead of assigning a unique neighbor from a neighborhood to each parallel worker, each model can be chosen with the same probability. For simplicity of calculations, we will consider below that N_k^o is implemented as Nr_k^o.

Definition Given a set of models M, and a model quality evaluation function $\psi : M \to \mathbb{R}$ we define a *performance-based distance* $d_\psi : M \times M \to N$ as follows. For every two models $m_i, m_j \in M, \psi(m_i) \leq \psi(m_j)$, let M_{ij} be the set of all models $m \in M$ such that $\psi(m_i) \leq \psi(m) < \psi(m_j)$. Then $d_\psi(m_i, m_j) = |M_{i,j}|$. We define that $d_\psi(m_i, m_j) = 0$ iff $i = j$.

Lemma 5.4.3 *Given $FullTop - k^l(\mathcal{M})$ with a graph G_{FT-k}, for which the probability distribution of reaching each model at level l, P^l is uniform, and an model quality function ψ, distance d. We impose a descending order, based on ψ, on $\{FullTop - k^l(\mathcal{M})\}$, $(\{FullTop - k^l(\mathcal{M})\})_{ord}$, so that $\psi(m_j) \geq \psi(m_{j+1}), \forall j = 1, \dots, \max - 1, \max = |\{FullTop - k^l(\mathcal{M})\}|$. Furthermore, this induces a descending order on the subset of $(\{FullTop - k^l(\mathcal{M})\})_{ord}$, $(\{(Nr_k^o)^l(\mathcal{M})\})$. Then for every three consecutive models in the ordered set $(\{(Nr_k^o)^l(\mathcal{M})\})$, $m_i, m_{i+1}, m_{i+2} \in (\{(Nr_k^o)^l(\mathcal{M})\}), i = 1, \dots, k-2$, on average, $d(m_i, m_{i+1}) = d(m_{i+1}, m_{i+2})$.*

Proof Let $max = min(k^l, \binom{n}{l})$. The number of all nodes at level l in the refinement graph G_{FT-k} is at most max. Let $X(1), \dots, X(k)$ be the random variables of the positions of the chosen models by $(Nr_k^o(\mathcal{M}^l))$ from all models in the ordered set $(FullTop - k^l(\mathcal{M}))$ in a decreasing order. For simplicity, among all the models in $\{FullTop - k^l(\mathcal{M})\}$, let m_0 be the optimal model, and m_{max} be the model with the worst performance. Then all three "gaps" $d(m_{X(j+1)}, m_{X(j)})$ as well as $d(m_{X(k)}, m_{max})$ and $d(m_0, m_{X(1)})$ have expected value of $\frac{max}{k+1}$. Let $E(X(1)|X(2))$ be the expectation of the event of selecting $X(1)$ which have already selected $X(2)$. Then $E(X(1)|X(2)) = X(2)/2$, because given $X(2) = m_i$, $X(1)$ is equally likely to be any of the models from $m_0, \dots, m_{i-1}$. Let $X(2) - X(1)$ denote the gap between the positions $X(1)$ and $X(2)$. Then $E(X(1)) = E(X(2) - X(1))$. Similarly, given $X(j) = i$ and $X(j+2) = p$, $m_{X(j+1)}$ has equal probability to be any of the models $m_{i+1}, \dots, m_{p-1}$. Thus, $E(X(j+2) - X(j+1)) = E(X(j+1) - X(j))$, $E(max - X(k)) = E(X(k) - X(k-1))$. It follows that all $k+1$ gaps have the same expected value. The total number of models at level l is max, so the expected value of the gap size is $\frac{max}{k+1}$.

5.4.1 The Graph, G_{FT-k}, Generated by $FullTop - k$.

Let us consider the graph that consists of the model subspace explored by $FullTop - k$ until refinement step l.

Definition Let G_{FT-k} be the graph generated by $FullTop - k$ exploring the space of models. Then the set of vertices V consists of the set of models explored by $FullTop - k$

until refinement level l. The set of edges E represents the relationship of direct refinement between the vertices. More precisely, $e = e(m_i, m_j) \in E \iff m_j \in r(m_i)$.

The graph G_{FT-k} is a subgraph of the search space graph $G_{\mathcal{M}}$.

Lemma 5.4.4 *The graph G_{FT-k} is a directed acyclic graph (DAG). Moreover, each node of G_{FT-k} has k out-degrees.*

Proof This is true by design.

Lemma 5.4.5 *The Nr_k^o Widening is equivalent to k independent random walks (performed by the parallel workers) on G_{FT-k}.*

Proof Follows by design of Nr_k^o. G_{FT-k} contains every potential choice of Nr_k^o and each parallel worker chooses exactly one node (model) at each step.

Lemma 5.4.6 *Let X, where $|X| = n$ be the set of model fragments, refinement operator of type* 1 *r and let $\mathcal{M}$ be the family of models, defined by r, X. The graph G_{FT-k} has at most* $\min(k^l, \binom{n}{l})$ *nodes at level l.*

Proof The number of models in $\mathcal{M}$ at refinement level l is at most $\binom{n}{l}$, while the number of different models in the refinement graph G_{FT-k} is at most k^l.

Probability Distribution Associated with G_{FT-k}

The solutions of $Nr_k^o(\mathcal{M})$ at level l depend on the structure of G_{FT-k}. Namely, it depends on the intersections between the refinement sets of selected models at each step in $FullTop - k$, or, equivalently, on how many *in-degrees* each model-vertex has. We know that at a given refinement level, each pair of refinement sets intersects at most once. This follows from the lattice structure.

Let P^l be the probability distribution for each node at level l to be discovered by a random walk. At each level l, the probability p_i^l for reaching a node m_i^l depends on the number of in-degrees to m_i^l as well as the probability distribution P^{l-1}. Let T be the transition matrix associated with G_{FT-k}.

Then,

$$P^l = P^{l-1}T.$$

This is demonstrated in Figure 5.4: the probability of reaching the purple, blue or yellow node is two times greater than the probability of reaching the red node. Let us consider several examples of Widening graphs and discuss the probability distribution

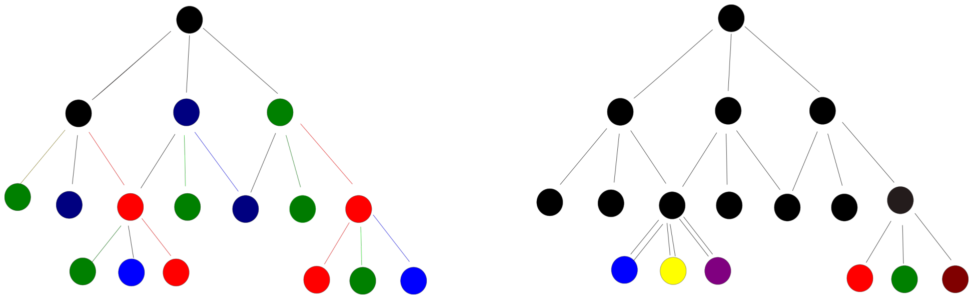


Figure 5.4: A figure representing the Widening via randomized neighborhoods as random walks on the graph of models.

of reaching their nodes. Figure 5.6 presents two extreme examples G_A and G_B, as well as a two more realistic examples G_C, G_D. In the first three examples, the probability distribution at level 3 is uniform. In graph G_D the probability distribution is strongly nonuniform. In these different cases of $FullTop - k$, Nr_k^o performs differently relative to $Top - k$. A uniform distribution P^l makes the solutions $\{Nr_k^o\}$ uniformly distributed across $\{G_{FT-k}\}$.

Uniformly structured graphs lead to uniform P^l, probability distribution of reaching the nodes at level l. Very degenerate refinement graphs, where P^l is strongly nonuniform, will make some nodes much easier to reach than others. The set of solutions will consists of repetitions of some solutions and others may not be at all reached, some regions from $\{G_{FT-k}\}$ will be overrepresented.

5.4.2 Ideal Widening Graph

Ideally, the Widening graph, G_W will have the following properties:

- Each node in the graph is reachable with equal probability.
- Each path of the graph passes through promising good solutions of the search space.
- At level l, the maximal pairwise diversity of the graph is equal to the maximal pairwise diversity of the search space at level l. Namely, $\Delta_{\max}(\{G_W^l\}) = \Delta_{\max}(\mathcal{M}^l)$.
- The minimal pairwise diversity is sufficient for the parallel resources not to be wasted on similar solutions, $\Delta_{\min}(\{G_W^l\}) > \delta$, for some appropriate δ.
- For a fixed Δ, the set of models, represented by nodes in $\{G_W^l\}$ is the set of models with the best Ψ among all sets of models, which satisfy the diversity requirements.

This is in fact a non-dominated set of solutions (part of the Pareto front) for the multi-objective optimization problem 3.3.

A graph with such properties will guarantee that random walks of sufficient number of parallel workers through such a graph will explore the important solutions of the search space. We will use *diversity and diverse neighborhoods* in order to achieve a Widening graph with good properties.

5.4.3 Uniform Distribution in G_{FT-k}.

The distribution of edges in G_{FT-k} determines the probability distribution of reaching each node of the graph. In Figure 5.6, G_B and G_C represent graphs with uniform probability distribution for reaching nodes at each level and Graph G_A has close to uniformly distributed intersections. In Section 5.4.6, we discuss the relationship between size of the neighborhood, k, and the probability distribution associated with the graph G_{FT-k}. Briefly, for smaller k it is more likely that the graph is degenerate, due to the higher chances of converging to local optima.

Lemma 5.4.7 *Given, P^l is uniform, the solutions discovered by Widening via Nr_k^o, $\{Nr_k^o(\mathcal{M})\} \in \{FullTop - k(\mathcal{M})\}$ are on average uniformly distributed among the solutions $\{FullTop - k(\mathcal{M})\}$. Thus $\max_\psi(\{(Nr_k^o)^l(\mathcal{M})\})$ will be on average at most $\frac{k^{l-1}}{2}$ models away with respect to the model quality function ψ from $\max_\psi(\{FullTop{-}k^l(\mathcal{M})\})$.*

Proof To begin with, let us consider each model discovered by $\{FullTop - k^l(\mathcal{M})\}$ as distinct. There are k^l models discovered by $FullTop - k$ at level l, $|\{FullTop - k^l(\mathcal{M})\}| = k^l$. Each of these models is reachable with equal probability by Nr_k^o, since each path traversed by $FullTop - k$ is equally likely to be traversed by Nr_k^o by design. So assuming k^l distinct models at level l in $\{FullTop - k^l(\mathcal{M})\}$, each of the models has equal probability of being chosen. From this follows that the k models discovered by Nr_o^k will be uniformly distributed among those k^l models of $\{FullTop{-}k^l(\mathcal{M})\}$. This implies that the $\max_\psi(\{(Nr_k^o)^l(\mathcal{M})\})$ will be at most k^{l-1} models away from $\max_\psi(\{FullTop - k^l(\mathcal{M})\})$.

We expect that degenerate Widening graph structure, such as in G_D from Figure 5.6, typically happens for a small k, when $FullTop - k$ is converging to some local optimum in the search space. However, the larger the number of parallel workers k the more the structure of G_{FT-k} is closer to the structure of the actual search space. We know that the search space forms a lattice and that each node for a given level l is reachable via l paths.

5.4.4 Widening Graph with Strongly Non-Uniform Distribution

This case can be related to the size of the neighborhood or a very unbalanced structure of the data. If k is small, then many of the neighborhoods can have among their optimal members the same models, which would represent converging to a local optimum. A greater neighborhood would improve that. An alternative way to improve the structure of the graph is through the use of *diversity*, as will be discussed in the later sections of this chapter. However, this can also be related simply to the graph structure of the particular search space and depends on the family of models.

Upper Bound for k

In the worst case, $max_{\psi}\{Top{-}\theta\} = max_{\psi}\{FullTop{-}\theta\}$. We want to have a quantitative estimation how large does k need to be, so that we can guarantee $\Psi(\{Top-\theta(\mathcal{M})^l\}) = \Psi(\{Nr^o_{\theta,k}(\mathcal{M})^l\})$ in the worst case scenario, where $\Psi(\{Top-\theta(\mathcal{M})^l\}) = \Psi(\{FullTop-\theta(\mathcal{M})^l\})$. In order to be able to guarantee that $\Psi(\{Nr^o_{\theta,k}(\mathcal{M})^l\})$ discovery of the best solution discovered by $Top-\theta$, k needs to be large enough to discover every solution at level l. This is related to the number of paths in $G_{FullTop-\theta}$ at level l. In $G^l_{FullTop-\theta}$ the models at level l are at most θ^l.

Lemma 5.4.8 *We assume an uniform distribution P of the edges in $G_{FullTop-\theta}$. For $k = \min(\theta^l, \binom{n}{l})$, $Nr^o_{\theta,k}$ explores fully the models explored by $FullTop-\theta$ at step l and guarantees $\Psi(Nr^o_{\theta,k}(\mathcal{M}))^l \geq \Psi(\{Top-\theta(\mathcal{M})^l\})$.*

Proof The graph $G_{FT-\theta}$ is a DAG, where each node has θ out-degrees. At level $l-1$ there is at most θ^{l-1} nodes, so the total number of edges will be at most θ^l. So for $k = \min(\theta^l, \binom{n}{l})$ we can guarantee that $\Psi_{\max}\{{N^o_{k,\theta}}^l\} = \Psi_{\max}(\{FullTop-\theta^l\})$.

In this worst case scenario, $Top-\theta$ discovers the best model from $\{FullTop-\theta^l\}$. Thus, a significantly larger number of parallel resources are needed for the communication-less Widening approach to be able to guarantee the same performance as that of $Top-k$ Widening. In order to avoid this degenerate situation there are several things to keep in mind. First, this worst case is based on the assumption that $Top-\theta$ does discover the optimal solution of the $FullTop-\theta$, which is an extreme scenario. Second, a small neighborhood size favors convergence to local optima, which is a big disadvantage especially for the communication-less method (although it is also a disadvantage of the $Top-k$, as it will explore different, but still similar solutions). Third, the utilization of diversity can help avoid degenerate graphs. We will discuss the properties of diverse neighborhoods later in this chapter.

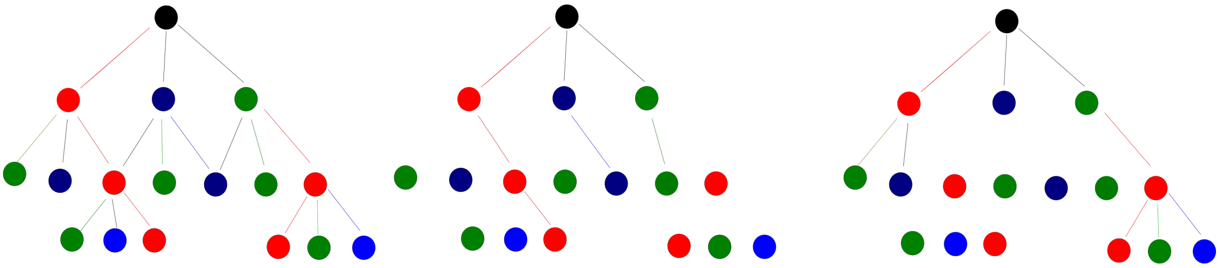


Figure 5.5: The artificial $FullTop-k$ structure, that bounds and contains both, N^o_k and $Top-k$

5.4.5 Upper bound for Performance of $Nr^o_{\theta,k}$ with Strongly Non-uniform Distribution P^l

Strongly non-uniform distribution is very disadvantageous for the N^o_k Widening methods in comparison to the $Top-k$ approach. In the case, where the distribution is strongly non-uniform, every model at level l needs to be reached, in order to be able to guarantee performance close to that of the $Top-k$ approach.

Lemma 5.4.9 *Assume that $P^l(x)$ represents the probability for each model at level l to be reached by a random walk on $G_{FullTop-\theta}$. Then, for $k = \frac{1}{\min P^l(x)}$ parallel random walks each model at level l will be reached on average.*

Proof For $k = \frac{1}{\min P^l(x)}$ on average the node reached by minimum number of paths will be reached.

5.4.6 Size of Neighborhood and Probability Distribution, P^l

Extremely degenerate graphs with strong intersections will more likely occur for small k. For large k intersections will be close to uniformly distributed, as they will be representing the structure of the search space. For small k these intersections represent getting stuck at a local peak. Of course all of this depends also on the general structure of the search space. We know that in our case the search space is a lattice with a known number of edges to each node at each refinement level. To deal with these degenerate graphs and prevent the parallel workers from converging to a local optimum, we need to use diverse neighborhoods, as they lead to a graph with properties closer to the ideal graph.

In the general case, the distribution of the solutions $\{Nr^{ol}_k(\mathcal{M})\}$ will depend on the distribution of the intersections of the neighborhoods, or the structure of the graph. If G_{FT-k} is degenerate as in Figure 5.6 D, the solutions discovered by the random walks will not be uniformly distributed. However, if the intersections are close to uniform, the

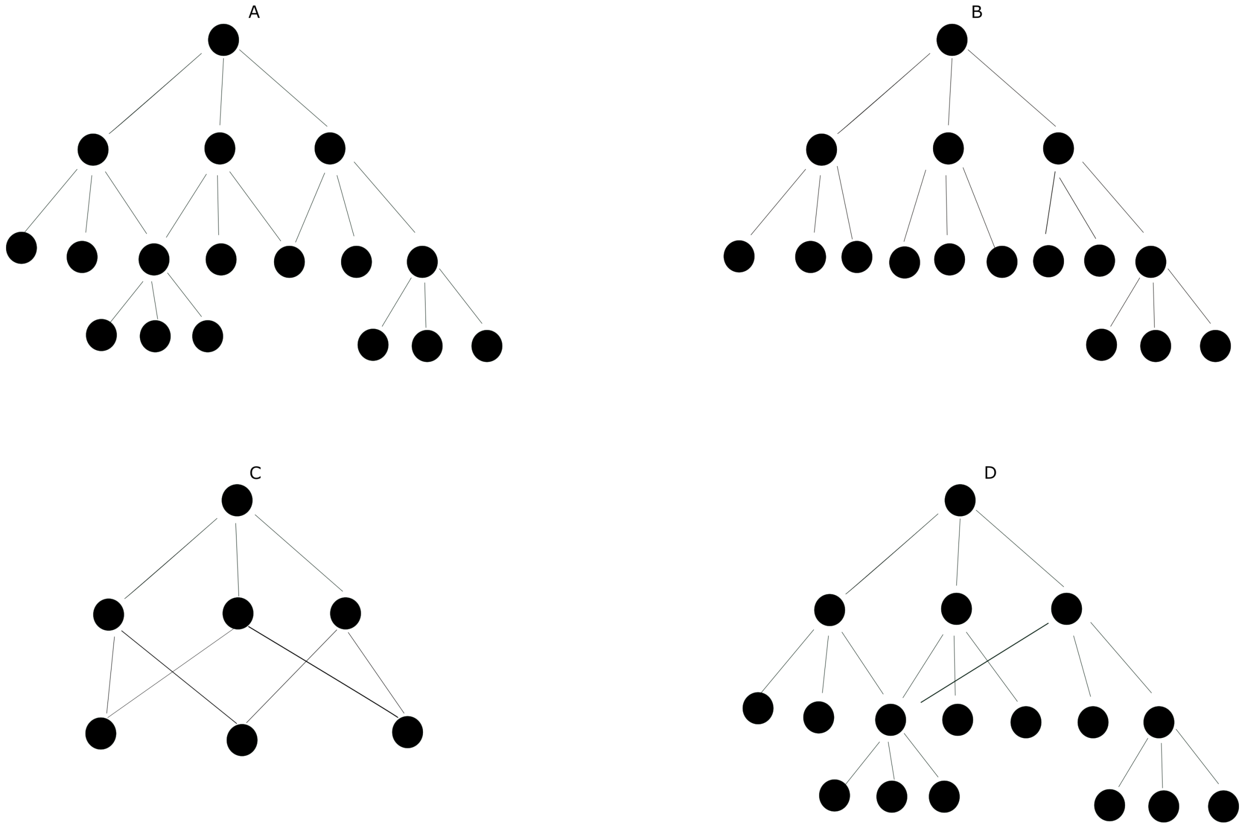


Figure 5.6: Examples of Widening graph structures with different probability distributions P^l.

solutions' distribution will also be close to uniform among the solutions of $\{FullTop-k(\mathcal{M})\}$.

By injecting diversity we can emulate a much larger neighborhood size and attempt to build a Widening graph with desired properties that spans promising and diverse regions in the search space in the search of the optimal solution.

We know that the refinement graph in the case of a simple refinement operator is a lattice, in which each node at level l can be reached via l paths.

Lemma 5.4.10 *Let* $\{FullTop-k^l(\mathcal{M})\} = \{m_1^l, \ldots, m_p^l\}$, *where* $m_i^l, i \in \{1, \ldots, k\}$ *are unique models, each repeated respectively* $n_1, n_2, \ldots, n_p$. *As* k *increases,* $n_1, n_2, \ldots, n_p \rightarrow n$.

Proof Follows from the lattice structure of the search space.

The k models discovered by Nr_o^k, will be uniformly distributed among those k^l models of $FullTop - k$. This implies that the $\max(\{(Nr_k^o)^l(\mathcal{M})\})$ will be at most k^{l-1} models away from $\max(\{FullTop - k^l(\mathcal{M})\})$.

If we want to improve this result and improve the distance from $\max(\{FullTop - k^l(\mathcal{M})\})$, we need to use θ, k-neighborhoods.

5.4.7 Properties of θ, k-Neighborhoods.

We can find actually how many parallel workers are needed for $\max_{score}\{(N_{\theta,k}^o)^l(\mathcal{M})\}$ to be some distance from $\max_{score}\{Top - \theta^l(\mathcal{M})\}$. Given the design of Nr_k^o, to improve its performance compared to $Top - k$, we need more parallel workers for a fixed size neighborhood.

Lemma 5.4.11 *Given a uniform distribution P^l, for $k = \theta^l/p$, the best solution discovered by $N_{\theta,k}^o$ is on average p models away from the best solution discovered by $FullTop - \theta$.*

Proof The solutions discovered by the parallel workers at step l, using $N_{\theta,k}^o$, $(\{N_{\theta,k}^o)^l(\mathcal{M})\}$ are uniformly distributed among the solutions discovered by $FullTop - \theta$. This follows from claim 5.4.7. $G_{FT-\theta}$ is the bound for $Top - \theta$, and thus $max\{N_{\theta,k}^o\}$ is at most p models away in the graph $G_{FT-\theta}$ from $\max_{\psi}\{FullTop - \theta\}$.

5.4.8 Conclusion

The flaws of this method are similar to the flaws of $Top - k$, they are related to lack of diversity among the solutions. However, due to lack of communication, the chances of obtaining similar solutions are greater. Strongly nonuniform intersections between the neighborhoods early on lead the search to focus on one area of the Widening graph. This is also a potential flaw of $Top - k$. Both approaches benefit from diversity, which helps to broaden the search and prevent the exploration of very similar solutions. In Section 5.6, we will demonstrate how diversity improves the property of the Widening graph by making the intersections rarer and more uniformly distributed, which would make the Widening graph one with desired properties. We will discuss both model-based diversity, which deals with the lattice structure, and data-based diversity, which deals with similarity/distance based on how models act on the data.

5.5 Properties of Widening via Similarity Neighborhoods

5.5.1 General Settings

Definition Let $d : \mathcal{M} \times \mathcal{M} \rightarrow \mathbb{R}$ be a metric defined on $\mathcal{M}$, let $(d, \mathcal{M})^n$ be a metric space of dimension n.

Based on this metric we will define a similarity k-neighborhood for each model $m \in \mathcal{M}$. In this section, we want to investigate the properties of Widening via similarity neighborhoods N_k^s. However, due to the different goals of this approach, we are interested in different aspects of the performance, compared to the optimality neighborhoods approach. N_k^s will assign to each parallel worker a different similarity neighbor. The goal of N_k^s is to explore some vicinity of the greedy solution with respect to a given similarity measure. Widening via this type of neighborhoods can be used in exploitation of the search, after a promising region is identified. A question arises, whether we can guarantee discovering the optimal solution within the area of exploration, or how far away will the solutions be from the optimal solution within the area of exploration? Let $M^{N_k^s} = \{m_1, \ldots, m_k\}$ be the final set of solutions of models of N_k^s. We are interested in how much these models diverge at level l. In order to answer this question, we need to know how the parallel workers implementing Widening via similarity neighborhoods behave in relation to one another. Namely, we are interested in what the size of the explored interval is, and in the distribution of the models, discovered by the parallel workers within this interval in relation to each other. We need to know two aspects: (i) the interval that Widening via similarity neighborhoods covers at level l, and (ii) the distribution of the models, reachable by N_k^s, within the interval. It is important to note that in a metric space, without a defined *norm*, there is no absolute position of a model, but only a pairwise distance between two models.

First, we will remind the following concepts of the metric spaces.

Definition A subset S of a metric space $(\mathcal{M}, d)$ is bounded if it is contained in a ball of finite radius, i.e. if there exists $m \in \mathcal{M}$ and $r > 0$ such that for all $m' \in S$, we have $d(m, m') < r$.

Taking advantage of the notion of similarity, we can assume that very similar models have very similar properties. Namely, for each small neighborhood of size δ, the models are of similar performance and their optimal refinements are similar. First, we will investigate the behavior of the models in the small neighborhood δ, in which we assume similarity of the properties leads to similarity of performance. Initially we will be interested in $\theta = \delta$.

Condition 5.5.1 *We shall assume that models within a small similarity neighborhood with radius $\frac{\delta}{2}$ have similar quality. Namely,*

$$d(m_i, m_j) \leq \delta \Rightarrow |\psi(m_i) - \psi(m_j)| < \lambda,$$

for a small λ.

Remark This assumption is not unrealistic, commonly for small enough δ the performance of the models is similar.

The next assumption limits the size of the neighborhood distance-wise for a very small number of neighbors.

Condition 5.5.2 *We shall assume that for every refinement set M^r, a neighborhood with diameter δ contains p models for some small number p, and assume that the distances of these models to the center of the neighborhood have some probability distribution $f(x)$.*

5.5.2 Refinement Graph $G_{N^s_{full}}$

Definition Let us consider the construct Nr^s_k, which assigns a parallel worker a model from the neighborhood of the locally optimal model uniformly at random.

For simplicity of calculations, we will discuss Nr^s_k below instead of $N^s_{k,k}$.

Definition We shall introduce the artificial construct N^s_{full}, which selects at each step all k similarity neighbors of a locally optimal model.

Definition We shall consider the graph $G_{N^s_{full}}$, for which the set of vertices V represent all models chosen by N^s_{full}, and the set of edges E represent the refinement relationship between the vertices.

Lemma 5.5.3 *$G_{N^s_{full}}$ is a DAG, where each vertex $v \in V$ has k out-degrees.*

Lemma 5.5.4 *Nr^s_k can be represented as k independent random walks on $G_{N^s_{full}}$.*

In order to evaluate k, the number of parallel workers needed so that we can guarantee that the best solution discovered by $\{Nr^s_k\}$ is at most a certain distance from the optimal model in an interval, we need to know the structure of the graph $G_{N^s_{full}}$. Namely, we need to know (i) how much $G_{N^s_{full}}$ diverges at a level l and (ii) the distribution of in-degrees for each subinterval δ of the interval I^l at level l. These two factors will determine the number of parallel resources needed in order to guarantee that the best solution discovered is at most δ from the best model in I^l.

For the Widening via similarity neighborhoods, the distance between the nodes matters. We are interested in the number of parallel workers we need to cover each subinterval of size δ from I^l. We are not only looking at in-degrees for each node from the graph, but also at total in-degrees towards nodes for each subinterval of size δ of I^l.

In order to evaluate how Nr^s_k behaves, we need to put (reasonable) restrictions on how much can the optimal refinements of two similar models differ. Namely, given two similar models neighbors $m_i, m_j : d(m_i, m_j) < \delta$, we assume that there is an upper bound on the distance between their optimal refinements, $d(m'_i, m'_j) < \epsilon$.

In particular we want to answer the question how many parallel workers are needed in order to guarantee that the best solution $m = max_\psi(M^{N^s_k})$ will have performance close to the performance of optimal solution $max_\psi(I^l)$.

Definition Given a family of models $\mathcal{M}$, $G_{N^s_{full}}(\mathcal{M})$ is the Widening graph for Widening via similarity neighborhoods N^s_k. We define I^l as the interval, determined by the models in $G^l_{N^s_{full}}(\mathcal{M})$. Namely, I^l is the largest pairwise distance of the models in $G^l_{N^s_{full}}(\mathcal{M})$.

5.5.3 Divergence of I^l

Condition 5.5.5 *We shall impose the following condition. For models within a small similarity neighborhood that* $0 \leq d = d(m_i, m_j) \leq \delta \Rightarrow d - \epsilon \leq d(m'_i, m'_j) \leq d + \epsilon$.

Lemma 5.5.6 *Given a size of similarity neighborhood δ, at step l, the distance between each two most dissimilar nodes of $G_{N^s_{full}}$ is at most $l(\delta + \epsilon)$.*

Proof We are interested in the size of I^l at a given level l, or, equivalently, how much each level of $G_{FullN^s_k}$ diverges. We will use the triangle inequality to investigate this.

Definition Given three points x, y, z in a metric space, the *triangle inequality* is defined as $d(x, y) \leq d(x, z) + d(y, z)$.

At level $l = 1$, consider a neighborhood $N_{\delta/2}(m_0)$ of the optimal model m_0S, i.e. all models which are at most $\frac{\delta}{2}$ from m_0. For each pair of models $m_i, m_j \in N_{\frac{\delta}{2}}(m_0)$, we find with the triangle inequality for the metric that the maximum distance between m_i and m_j is δ, $d(m_0, m_i) \leq \frac{\delta}{2}, d(m_0, m_j) \leq \frac{\delta}{2} \Rightarrow d(m_i, m_j) \leq \delta$.

At level $l = 2$, again following from the triangle inequality, and from Condition 5.5.5 the maximal distance between a pair of models m^2_i, m^2_j is $d(m^2_i, m^2_j) \leq \epsilon + 2\delta$.

It follows that at step l the interval I^l will be bound by $l(\delta + \epsilon) - \epsilon$.

While the metric space is sufficient to find an upper bound for the divergence of the interval I^l at a step l, in order to calculate the distribution of the paths towards different subintervals in the search space, we need to specify additional properties of the space, because probability distributions look differently in different types of spaces.

In order to evaluate more concretely the number of parallel workers needed, we will add more specific requirements on the space $(\mathcal{M}, d)$. In the literature neighborhoods of similar points are often modeled either by uniform or Gaussian distribution.

We will discuss one particular examples of normed spaces of models, which are applicable in our case– Euclidean space.

5.5.4 General Space

Given that in each neighborhood of size δ the probability distribution of the distance of models from the center of the neighborhood is $f(x)$, then at level l the probability distribution of the distance of models from the center of the explored interval I will be characterized by the convolution:

$$P^l(x) = (P^{l-1} \circledast f)(x).$$

In the general case, for any space, it is not clear how this convolution looks like, it will depend on the properties of the solution space.

Most of the algorithms generate a space, which is discrete. In such a situation, the distribution of models within the dispersion interval I^l is calculated in the same fashion as above.

The resulting probability distribution from the convolution is going to depend on the properties of the hypothesis space. We will discuss an example using Euclidean space for demonstration purposes, due to the convenience of calculations. This space is not a good representation of the typical model space, which we are discussing. There are many types of machine learning problems for which the models in the search space can be presented as points in a discrete subspace of the Euclidean space or in the Euclidean space. The hypothesis space of the algorithm *backpropagation* is the continuous Euclidean space. Another example, similarity mining of association rules can use Euclidean distance as a metric on the vector bit encoding of the rules, confidence or other properties[134] . In this situation, Euclidean distance is not optimal, because it does not capture a lot information, but is commonly used.

5.5.5 Illustrative Example: Euclidean Space

For simplicity of calculations in this illustration, we will investigate how many parallel workers are needed to cover I^l in the context of Euclidean spaces, not just metric spaces.

We will put additional restrictions for $(\mathcal{M}, d)$. Namely, we will assume that the models from $\mathcal{M}$ can be presented as points in Euclidean space, and that distance d is the Euclidean distance. Recall the definitions of Euclidean norm and Euclidean distance. Euclidean norm is defined as

$$\|\mathbf{x}\| = \sqrt{\mathbf{x} \cdot \mathbf{x}} = \left(\sum_{i=1}^{n} x_i^2 \right)^{1/2}. \tag{5.1}$$

And the Euclidean distance between two vectors $\mathbf{x} = (x_1, x_2, ..., x_n), \mathbf{y} = (y_1, y_2, ..., y_n) \in \mathbb{R}^n$ is defined as

$$\|\mathbf{x} - \mathbf{y}\| = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + ... + (x_n - y_n)^2}. \tag{5.2}$$

For illustration we will take a simple example. Consider that the family of models $\mathcal{M}$ consists of simple rules of the type

$$x_1 = v_1 \wedge x_2 = v_2 \ldots \wedge x_i = v_i \rightarrow d.$$

The set of all possible attributes is A, $|A| = n$, and each attribute is numerical with domain $\mathbb{R}$. Then each rule can be presented as a vector in an Euclidean space $\mathbb{R}^n$, based on the values for each attribute, where the number of dimensions n is the total number possible attributes for the rules. Then, the similarity between two rules is evaluated based on the values of each attribute.

Each model is a n-dimensional vector, where for position i the value is either 0, if this attribute is not added by the refinement operator, or is the value of the attribute, which is added by the refinement operator. To generate a new refinement of a model m, the refinement operator adds a model fragment, which is the couple $(attribute, value)$.

First, let us look at the general case, where we impose no restrictions on the probability distribution of models in a neighborhood of diameter δ. We will see how we can obtain the distribution of the models at level l.

Lemma 5.5.7 *We shall assume a (continuous) distribution $K(x, x')$ for a fixed x' is the distribution of models in the neighborhood of diameter δ of the optimal choice of the model at x' and P^l be the probability distribution of models (nodes of the graph $G_{N^s full}$) at level l.*

Then

$$P^{l+1}(x) = \int K(x, x') P^l(x') d^n x'.$$

Second, let us consider a more specific example. Let us assume that the models in each neighborhood of diameter δ follow a different Gaussian distribution.

Condition 5.5.8 *We shall assume that each small neighborhood of similar models* $\delta(x)$ *follows a different Gaussian distribution* $G(\Sigma(x'), M(x'))(x)$, *where* M *is the center of* δ *and* Σ *is the standard deviation of the Gaussian.*

Then in the continuous case, we will calculate the distribution of models at level l as follows.

Lemma 5.5.9 *We impose Condition 5.5.8, then*

$$P^{l+1}(x) = \int P^l(x')G(\Sigma(x'), M(x'))(x)d^n x'.$$

Note, if we want to model the neighborhood distribution in a discrete fashion, the same principle applies. Instead of using integration, in this case a summation of the products of the two functions is used.

Lemma 5.5.10 *We assume Condition 5.5.8*

$$P^{l+1}(x_n) = \sum_{n'} G_{nn'} P^l(x_{n'})$$

x_n *is the discrete coordinate of the model,* $G_{nn'}$ *is a Gaussian for the coordinate* $x_n - M(x_{n'})$ *with standard deviation* $\Sigma_{n'}$, $G_{nn'} = G_{\Sigma_{n'}}(x_n - M(x_{n'}))$.

For simplicity we assume that each neighborhood follows the same Gaussian distribution $G_\delta(\Sigma, M)$.

Condition 5.5.11 *We shall assume that models within a neighborhood of size* δ *follow a Gaussian distribution* $G_\delta(\Sigma, M)$, *where* $M = (0, \dots, 0)$, $\Sigma = \left(\frac{\delta}{6}; \dots; \frac{\delta}{6}\right)$.

In this scenario we can apply *convolution* of two density functions multiple times in order to calculate the probability density function P^l for the distribution of models in I^l.

Definition Given two functions f and g, a convolution of the two functions over $\mathbb{R}^n$

$$(f \circledast g)(x) = \int f(x - x')g(x')d^n x'$$

Recall the definition of a Gaussian function.

Definition A Gaussian function is defined as

$$G(\Sigma, M)(x) = \frac{1}{\sqrt{2\pi\Sigma^2}} e^{-\frac{(x-M)^2}{2\Sigma^2}}.$$

The Gaussian function has very convenient convolution properties. We remind the following known fact.

Fact 5.5.12 *A convolution of two Gaussian functions, $G_1(\Sigma_1, M_1)$ and $G_2(\Sigma_2, M_2)$ is a Gaussian $G_{G_1 \circledast G_2}(\Sigma = \sqrt{\Sigma_1^2 + \Sigma_2^2}, M = M_1 + M_2)$.*

Lemma 5.5.13 *We impose Condition 5.5.11. The distribution of the models at level l is*

$$P^l = P^{l-1} \circledast G(\Sigma, 0).$$

P^l is a Gaussian with a standard deviation $\sqrt{l}\Sigma$, $P^l = G(\sqrt{l}\Sigma, 0)$.

Proof The probability distribution at level l depends on the probability density of the previous refinement level and the distribution of models in each similarity neighborhood at level l. This is calculated as the convolution of the two functions.

$$P^l = \int P^{l-1}(x) G(\Sigma, M)(x) dx.$$

At step l $\Sigma_l = \sqrt{l}\Sigma$, this means

$$P^l = G(\Sigma_l, M_l) = G(\sqrt{l}\Sigma, 0).$$

For simplicity, we model the distribution of the models in the small neighborhoods of diameter δ in a continuous way. However, it can also be modeled using a discrete distribution, such as the *uniform* distribution, which can be more appropriate. The resulting distribution P^l, however, will not be drastically different. If the small δ neighborhood is modeled by discrete uniform distribution, then via the central limit theorem, for a large l the convolution of $(P^l \circledast f)(n)$ will approximate a Gaussian distribution. In the general case this is true for any distribution with finite moments. How many parallel workers do we need to cover the whole interval I^l? Clearly, due to the fact that P^l is a Gaussian, the central part of the interval I^l will be very easy to cover and will require a small number of parallel workers and in order to cover the periphery of the interval I^l a much larger number of parallel workers.

The number of parallel workers needed to cover the interval I^l given distribution $P^l = G(M^l, \Sigma^l)$ will depend on $\Sigma^l = \sqrt{l}\Sigma$. We know that $\frac{\delta}{2} = 3\Sigma$ and that $I = l\delta = 6l\Sigma$. With this in mind, how many parallel workers do we need to cover I in such a way that there is no interval of size δ without at least one model? Within two standard deviations the number of δ neighborhoods is $\lfloor \frac{\sqrt{l}}{3} \rfloor^n$. Each neighborhood δ will be reached with a different probability, based on the Gaussian. The closest models to the mean m_0 will be the easiest to reach. Let

$$P(x_1) = 1/p_{\delta_1}, P(x_2) = 1/p_{\delta_2}, \ldots, P(x_q) = 1/p_{\delta_q},$$

where $P(x_i) = 1/p_{\delta_i}$ is the probability to reach the i-th neighborhood of diameter δ from the mean m_0.

Theorem 5.5.14 *Notation as above, let $P^l = G(M^l, \Sigma^l)$ be the Gaussian distribution of paths to nodes, and let the interval $I^l = \sqrt{l}\delta = 6\sqrt{l}\Sigma$. Then the number k of parallel workers needed to cover each δ-sized subinterval of the interval I^l is*

$$k = p_q,$$

where $q = \lfloor \frac{\sqrt{l}}{3} \rfloor$.

Proof Let X be the event of having a parallel worker in each interval with diameter δ, starting from the mean of the Gaussian (the center of I^l), and then going towards the edges of I^l and let $\delta_1, \delta_2, \ldots, \delta_l$ be the subintervals starting from m_0. Each subinterval, depending on its Z score has a different probability to be visited by a random walk with δ_1 having the highest, and the farther the subintervals from the center, the more parallel workers are needed in order to have at least one model visited in that interval. In order to guarantee that each δ neighborhood is visited, we need $k = p_{\delta_{l-1}} = p_{\delta_l}$, where δ_{l-1}, δ_l are the neighborhoods, which are farthest removed from m_0.

Based on our goals, we can estimate how many parallel workers are needed to cover different parts of the interval I^l. Different δ-neighborhoods will require a different number of parallel workers based on their Z score,

$$Z = \frac{X - M^l}{\Sigma^l}.$$

We know that $\Sigma^l = \sqrt{l}\Sigma$.

For example, consider the subinterval of I^l, that is 1 standard deviation away from the mean. The size of that subinterval is $2\sqrt{l}\Sigma$. If we want to guarantee that at least one model will be distance no greater than δ similarity-wise from the best model in the region, we need $k = p_{\frac{\sqrt{l}}{3}}$ random walks, where

$$\frac{1}{p_{\frac{\sqrt{l}}{3}}}$$

is the probability to reach the most distant δ-neighborhoods within the subinterval of size $2\Sigma^l$, the edges of which are Σ^l away from the mean.

5.5.6 Shift ϵ

Condition 5.5.15 *We shall impose that ϵ follows a Gaussian distribution $G_\epsilon(M, \Sigma)$, where $M = \begin{pmatrix} 0 \\ \ldots \\ 0 \end{pmatrix}$, $\Sigma = \begin{pmatrix} \epsilon \\ \ldots \\ \epsilon \end{pmatrix}$.*

Lemma 5.5.16 *We impose Conditions 5.5.5,5.5.11, 5.5.15. Then the probability distribution P^l at level l is the convolution:*

$$P^l = P^{l-1} \circledast G_\delta \circledast G_\epsilon.$$

Compared to the simple modeling without such shift, in this case the convolution will result in a wider Gaussian for P^l, which means that the Σ^l will be greater. Then more workers will be required to cover the center of the interval I^l, since the density will decrease, compared to the previous scenario, but it will be easier to cover the subintervals of I, which are away from the center. On an intuitive level: if the paths from the center shift towards the border of the interval, then the distribution of the paths would be wider and a larger part of the interval I^l would be covered with fewer parallel workers. However, with large shifts ϵ there is a danger that there is no coverage of I^l by N_k^s, because in reality there are a fixed number of models at each level, and this aspect cannot be captured by using continuous functions. We will discuss this below.

Coverage of I^l

In the previous section, we modeled the distribution of the models using a Gaussian including the shift ϵ in order to see how the probability distribution P^l changes as the search progresses.

In reality, for some model families the search space is discrete. For others, such as neural networks, it is continuous. If the search space is continuous, then regardless of the shift ϵ, there will always be a large number of parallel workers, which will cover the interval I^l.

For discrete spaces, however, if there is a very large shift ϵ, which happens only in one model and its refinements, it can happen that $\{N_k^s\}$ does not cover I. The modeling via a Gaussian does not capture such a situation. If there is an uncovered area in I, regardless of how large is k, Widening via similarity neighborhoods is not able to guarantee performance close to that of the best model from the interval I^l.

Lemma 5.5.17 *Given that there is coverage of I^l, regardless of P^l, there exists a large enough k for which the Widening approach performs as well as the best model in I^l.*

Proof Follows from the definition of coverage.

Lemma 5.5.18 *Assume Condition 5.5.2 holds. Then, for $\epsilon < \frac{\delta}{2} - \frac{\delta}{p}$, I^l will be covered by $\{{N_{full}^s}^l\}$ for any l.*

Proof By assumptions 5.5.5 if $\epsilon < \frac{\delta}{2} - \frac{\delta}{p}$, two neighbors at level $l = 2$ cannot diverge more than ϵ and thus will have at least one model m as a common model. At level

l these models/this model will have a neighborhood that is covering and the out-most members of this neighborhood will be at most $\epsilon < \frac{\delta}{2} - \frac{\delta}{p}$ from the descendants of their nearest members.

These numbers are based on our particular fixed assumptions and settings, but if ϵ is relatively small compared to the size of the neighborhood there will be coverage of I^l, otherwise it cannot be guaranteed.

Note: This holds only if ϵ is small enough to guarantee coverage at level l by the nodes of the graph $G_{N^s_{full}}$, $\{G^l_{N^s_{full}}\}$.

An Argument for Small ϵ

For two similar models a small value for the shift ϵ is to be expected, especially later in the search. In a single refinement step only a small part of the model is changed, so this means that all refinements of two similarity neighbors are not too different. Intuitively, the optimal refinements of the two very similar models will be similar as well. This depends on the particular data and search space and will not hold universally.

5.5.7 Conclusion

Widening via similarity neighborhoods is a parallel search that is focused on a particular area of the space of models $\mathcal{M}$. Following the assumptions, that similar models have similar properties, this method investigates models similar to the greedy choice. We use the assumption that similar models behave similarly, that their performance is similar, as well as, that optimal refinements of similar models are not dissimilar. This leads to a predicable behavior in which the center of the interval I^l has much higher density of selection paths compared to the edges of the interval I^l, which is due to many more intersecting neighborhoods in the center compared to the number of neighborhoods towards the boundaries. In order to cover the more central parts of I^l much less parallel workers are needed, compared to the resources needed to investigate the edges of I^l.

5.6 Properties of Widening via Diverse Neighborhoods

The weakness of Widening via optimality neighborhoods is similar to that of $Top-k$ Widening, albeit potentially more exaggerated due to the lack of communication, which prevents the removal of duplicated models. As already discussed, $Top-k$ Widening has the potential weakness of converging by exploring only similar solutions and needs diversity to be enforced.

As we discussed the Widening via optimality neighborhoods can be represented as k random walks on the graph of $FullTop{-}k$. Thus its behavior depends on the structure of this graph. If the distribution of the number of paths reaching a given model is strongly non-uniform, the behavior of the random walks will also be non-uniform.

The goal of Widening via diverse neighborhoods is to force the parallel workers to explore diverse and promising parts of the search space in a way that does not require communication between the parallel workers. Both, model quality and model diversity have to be taken into account when generating diverse neighborhoods. We show that by using diversity we build a graph closer to the *ideal Widening graph*, compared to the Widening graph G_{FT-k}. By applying diversity approaches, we intend to build a Widening graph that spans all important high-quality regions of the search space graph and discover the optimal solutions in those regions.

5.7 Types of Approaches to Diversity

In this section, we will systematize the different approaches to diversity that one could take and discuss their disadvantages and advantages.

5.7.1 Model-based versus Data-based Diversity

Model-based diversity is evaluated based on the model fragments used by each model, data is not taken into consideration when evaluating this type of diversity. Model-based diversity deals with the distance between two models within the refinement graph G.

On the other hand, in the approaches that use data-based metrics, the distance between the models is evaluated based on how the model acts on the data. Given the refinement operator is of type 1, for the models in a fixed refinement set, each pair of models has the same model-based distance, since they differ in one model fragment. However, they may have different data-based distances.

5.7.2 Global versus Local Diversity

A local approach to implementing diversity deals with enforcing diversity only within each neighborhood. In contrast, the global diversity approaches use such an approach to diversity that enforces it globally for the parallel workers.

An example of local diversity is to enforce a diverse selection in each refinement set. Regardless what the models are, from a given refinement set the parallel workers are assigned a diverse subset of models. However, if one parallel worker is choosing from one refinement set and another parallel worker, via its refinement path is choosing from a different refinement set, they can choose the same model, or very similar models. Widening using diverse neighborhoods is a local approach. The strength of the local diversity approach is that it is easy to implement in a communication-less fashion, while following the greedy logic of the algorithm and taking into consideration the performance of the models selected at each step. The main issue with regards to local diversity approaches is: **Does the locally diverse behavior lead to the parallel workers exploring different parts of the model search space globally?**

A global approach to diversity defines such a behavior that guarantees that globally each parallel worker will choose a different model no matter what. In this approach the problem is rigidity and the fact that due to this rigidity it is difficult to incorporate the performance of the models together with the diversity. Examples of a global approach include assigning preferences directly to certain model fragments, i.e. each parallel worker prefers certain model fragments, ban lists on certain model fragments, i.e. that some workers are not allowed to choose certain model fragments, preferences to particular data points, and others. The difficulty of this approach is assigning such preferences or bans in a good way.

5.7.3 Local Approach using Data-based Diversity. Widening via Diverse Neighborhoods

Note 5.7.1 ***Widening via diverse neighborhoods is possible only using data-based diversity, because the diverse neighborhoods are built only from models from the same refinement set, which differ by just one refinement.***

Note 5.7.2 ***For models, built by the refinement operator of type* 1*, the data-based diversity between the model refinements (or the respective model fragments) needs to be recalculated at every step.***

There are two important aspects we need to investigate when it comes to the properties of Widening via diverse neighborhoods. The first is: **Does the local use of diversity lead to globally diverse behavior?** Namely, does the diversity in neighborhoods improves the search space exploration in comparison with Widening via optimality neighborhoods. The second is: **How does Widening via diverse neighborhoods, a communication-less method, compare with diverse $Top-k$?**

Let d be a *data-based* metric, and $(\mathcal{M}, d)$ be the metric space, defined by the metric d and the family of models $\mathcal{M}$. As discussed in Chapter 3, the models selected are part of the Pareto front, consisting of the best solutions of the multi-objective optimization problem described in 3.3. The diverse k-neighborhood N_k^d of a model m is formed by selecting the k best diverse models, a Pareto non-dominated set from a refinement set $M^r = r(m)$. At both extremes of dominant solutions are the k models with highest optimality, which maximizes $\Psi(m_1, \ldots, m_k)$ and the k most diverse models, which maximize $\Delta(m'_1, \ldots, m'_k)$. Typically, it is beneficial to use sets of solutions that do not fall in both extremes of the front. Depending on the structure of the search space, different strategies can be the most beneficial: ones that give more importance to the optimality, and those which give more importance to the diversity. The relative importance of diversity versus optimality is difficult to determine apriori, the optimal strategy will depend on the landscape of the model space. But this is the issue that the Widening methods, with communication also face.

We define $Top - d, k$, the diverse version of $Top - k$ as an approach, which, given models $m_1, \dots, m_k$ selected at step $l - 1$, at level l selects a non-dominated set with respect to diversity and optimality, of k models from $\cup\{r(m^1), \dots, r(m^k)\}$.

Definition Given that a set of k models $\{m_1, \dots, m_k\}$ is selected at level $l-1$, then $Top-d, k$ selects at level l the non-dominated set of k models $\{m'_1, \dots, m'_k\}$ from $\cup_{i=1}^{k} r(m_i)$, from the Pareto front of multi-objective problem, described in 3.3.

In contrast, N_k^d builds a non-dominated set of k models for each refinement set $r(m_i), 1 \leq i \leq k$, because it is a communication-less approach. Below we give a formal definition.

Definition Let $\mathcal{M}$ be a family of models, let r be a refinement operator of type 1 over $\mathcal{M}$, let d be a data-based metric. We introduce the diverse k-neighborhood N_d^k of a model $m \in \mathcal{M}$ as follows: $N_k^d = \{m'_1, \dots, m'_k\}$, where $m'_i \in r(m), 1 \leq i \leq k$, where the set $\{m'_1, \dots, m'_k\}$ is a non-dominated solution set with respect to diversity and optimality, for the problem described in 3.3.

5.7.4 Pareto Front, Types of Diverse Neighborhoods

The diverse neighborhoods N_k^d can be built using different approaches. The non-dominated set of solutions can be chosen in various ways, e.g. via simple threshold, or by using methods from genetic algorithms, as described in the previous chapter. Depending on the goals of the search and the structure of $(\mathcal{M}, d)$, different approaches will be beneficial.

We describe two examples of Widening via diverse neighborhoods which differ in the way the non-dominated set is selected from a given refinement set M^r.

The Widening approach N_k^δ chooses from a given refinement set the k optimal models that are at least a distance δ from each other.

The Widening approach N_k^{lp} is focused on detecting the k best local peaks. A model m belongs to the k dominant solutions, if it is a local peak, i.e. it has among the k-th largest distances to the model that is of the same or better quality.

We also discussed that approaches, inspired by genetic algorithms, such as crowding and fitness sharing, are also used for peak detection, and choosing models from different sub-populations, taking into account both, diversity and optimality.

5.7.5 Widening Graph of Diverse Neighborhoods.

Definition Let $\mathcal{M}$ be a family of models, r a refinement operator over $\mathcal{M}$, and N_k^d a Widening approach via diverse neighborhoods. We introduce the artificial construct $FullN_k^d$, which at each step selects all the models which could potentially be selected by N_k^d. We will define N_k^d inductively. For the base model m_0, $N_k^d(r(m_0)) \subset FullN_k^d(\mathcal{M})$. For $m_i \in \{FullN_k^d(\mathcal{M})\}$, $N_k^d(r(m_i)) \subset FullN_k^d(\mathcal{M})$.

Definition Let m be a model in the family of models $\mathcal{M}$, with a refinement operator r. Nr_k^d is an operator that defines randomized diverse neighborhood Widening, which assigns a model from$\{N_k^d(r(m))\}$ to each worker at random, with equal probability.

Lemma 5.7.3 $Nr_k^d(\mathcal{M}) \subset FullN_k^d(\mathcal{M})$, $Top - d, k(\mathcal{M}) \not\subset FullN_k^d(\mathcal{M})$.

Proof The fact that $\{Nr_k^d\} \subset \{FullN_k^d\}$ follows by definition. We will show that there can exist a model m, $m \in \{Top - d, k\}^l$, $m \notin \{FullN_k^d\}^l$. If the model $m \in \{Top - d, k\}$, this implies that m is a member of the non-dominated set of solutions from $\cup M_i^r, i \in \{1, \ldots, k\}$. Let the model m be a member of the refinement set M_j^r for a fixed $1 \leq j \leq k$. However, the model $m \in M_j^r$ may not be a member of the local non-dominated solution on M_j^r, $m \notin N_k^d(M_j^r)$. If a model $m \in Top - d, k(\cup M_i^r)$ is not selected by N_k^d, this means that there is another model m' in the same refinement set(s) as m, so that $\psi(m') > \psi(m)$. This model m' is dominant locally, but is not dominant globally. This can be only due to diversity. For a set of models M_i^r, for a fixed $1 \leq i \leq k$, $m, m' \in M_i^r$ there is another model $m'' : \psi(m'') \geq \psi(m'), d(m'', m') < \delta$.

Issues with Widening via Diverse Neighborhoods

The problem of Widening with diversity neighborhoods stems from the use of diversity locally, only within a given refinement set, without communication between the parallel workers. There is no information about the full landscape at a refinement level, and because of the use only of local refinement set information, each can potentially select the same or similar diverse models for their neighborhoods. Consider a local peak $m_{lo} \notin \cup N_k^d(M_i^r)$, but $m_{lo} \in Top - d, k(\cup M_i^r)$. This can happen only if models for given refinement sets are better than m_{lo}, however to be selected globally this peak must be one of the optimal peaks in the respective refinement set as well, thus there are peaks in other refinement subsets that dominate (are better in performance and close to) over the preferred peaks. The only way local approaches can handle such models that are too similar, but are from different refinement sets/neighborhoods, is to remove their predecessors at a previous refinement step.

Definition We define the graph $G_{FullN_k^d}$, for which the set of vertices V represent the models visited by $FullN_k^d$ and the edges E represent the refinement paths between these models.

Lemma 5.7.4 *Nr^d_k is equivalent to k random walks on $G_{FullN^d_k}$.*

Proof Follows from the definitions of both Nr^d_k and $G_{FullN^d_k}$.

Diversity and Size of Neighborhoods

We can see in the experimental chapters, Chapter 7 and Chapter 8, that just by increasing the size of the neighborhood, we do not get improved solutions. There are two aspects to the size of neighborhoods in Widening via neighborhoods. First, simple randomization does not necessarily imply improvement of diversity, due to not knowing the background distribution of the models. It is necessary to enforce diversity directly, when selecting models. Second, just relying on diversity does not lead to exploration of the interesting areas in the search space and to a good set of final solutions. This is because if there is no requirement for optimality, but just for diversity, this does not lead to solutions with good quality.

5.7.6 The Goal of Diverse Neighborhood Widening

Even though the full search space exploration will guarantee the discovery of the optimal model, it is enormous and typically the vast majority of models in such a search space consists of "bad" and "unimportant" models. Thus it is usually not practical to do such full exploration.

Utilizing diversity, with or without communication, is a way to *explore* the (potentially) meaningful parts of the search space and approximate the results of a full exploration, by carefully selecting which parts of the search space to explore. Diversity-driven exploration is, in essence, preserving paths in the search space that contain important information and pruning the refinement paths that contain no important information. This includes poorly performing models, as well as, models that are highly similar to each other. At level l, Widening methods with and without communication aim to preserve the important information of $\mathcal{M}^l$ – the models, which are diverse and perform well. Both methods, $Top - d, k$, the one with communication, and N^d_k, the one without, aim to find the k most "important" models of $\cup(M^r_i)^l, 1 \leq i \leq k$. However, N^d_k has only access to each refinement set separately, and has to choose the dominant models locally from each set, which adds an additional restriction. Both methods work under the assumption that models that perform well at level $l - 1$ will have refinements that perform well at level l and that models that are similar at level $l - 1$ will have refinements that are similar at level l.

Neither of these methods can guarantee the selection of the non-dominated set of models at each refinement level of $\mathcal{M}^l$ for arbitrary k. However, $Top - d, k$ performs better due to building a non-dominated set of temporary solutions over $\cup r(m_i), i =$

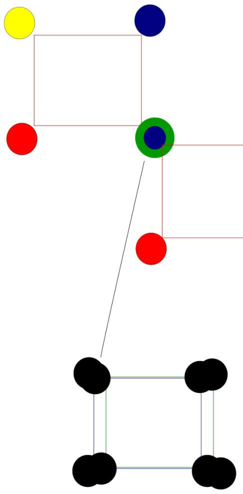


Figure 5.7: A figure representing an intersection of two diverse neighborhoods in a single model. At the next step the two parallel workers, which are considering this model will produce two identical diverse neighborhoods. Because these neighborhoods are diverse, their respective members are different from the other members of the neighborhood. Thus the repeated models (which are members of two identical diverse neighborhoods) will be dispersed and not clustered in a single small part of the search space, which is what can potentially happen in Widening via optimality non-diverse neighborhoods.

$1, \ldots, k$, while N_k^d is building non-dominated solution sets locally, in each refinement set. The latter approach can lead to selecting a set of solutions which are locally, in every neighborhood, non-dominated, but are not a part of the Pareto front in the union of the refinement sets.

Diversity-driven Widening with or without communication can only work under the assumption that very similar models will have similar refinement sets, while very diverse models will have diverse refinement sets. By removing very similar models at levels $1, \ldots, l-1$ in diversity-driven Widening, the potentially very similar refinements of these models are removed at level l.

Condition 5.7.5 *If $d(m_i, m_j) \gg \xi$ for some large ξ, then $\Delta(r(m_i), r(m_j)) \gg \xi$. Models, which are very different, will have very different direct refinements. Models which are very similar will have very similar direct refinements. If $d(m_i, m_j) \leq \epsilon$ for some small ϵ, then the pairwise diversity of the set $\Delta(r(m_i), r(m_j)) \leq \sigma$ for a small σ.*

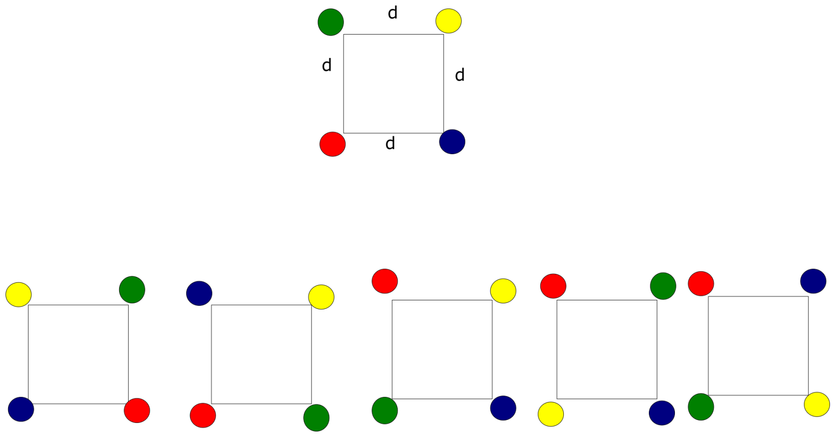


Figure 5.8: A figure representing an idealized graph G_{N^d}. The graph consists of diverse, nonintersecting neighborhoods of non-dominated sets of solutions.

5.7.7 The Effect of Using Local Diversity within Neighborhoods on the Global Diversity of the Search

The ideal scenario, when diverse neighborhoods are used is shown in Figure 5.8, which shows the ideal graph for Widening via diverse neighborhoods. There diverse neighborhoods which do not intersect with each other are built in each step. However, this is not realistic, because there is no communication between the workers, and the diversity used is *local diversity*.

The realistic scenario for Widening via diverse neighborhoods, is to assume there will be intersections, but that these intersections will be fewer than for Widening via optimality neighborhoods, and will be more uniformly distributed, in comparison to Widening via optimality neighborhoods. This graph is shown in Figure 5.9.

In order for the local diversity, used by a single parallel worker within a refinement set, to affect the diversity of the search, we need to impose some conditions (it is not true in the general case). Namely, we need assumptions with regards to the set diversity, Δ, of the refinement sets of two models, and the corresponding diverse neighborhoods, defined on those sets.

Condition 5.7.6 *Let* $m_i, m_j \in \mathcal{M}$*. Then* $\Delta(r(m_i), r(m_j)) \propto d(m_i, m_j)$*.*

Condition 5.7.7 *Let* $m_i, m_j \in \mathcal{M}$*. Then* $\Delta(N_d^k(r(m_i)), N_d^k(r(m_j))) \propto d(m_i, m_j)$*.*

Only under such conditions, which entail that for very different models their respective refinement sets and consecutively their diverse neighborhoods will be different, we can claim that local diversity is translated into diversity of the solution, $\Delta_{max}(\{FullN_k^{\delta}(\mathcal{M}^l)\})$ at each level and it will be better than not implementing diversity.

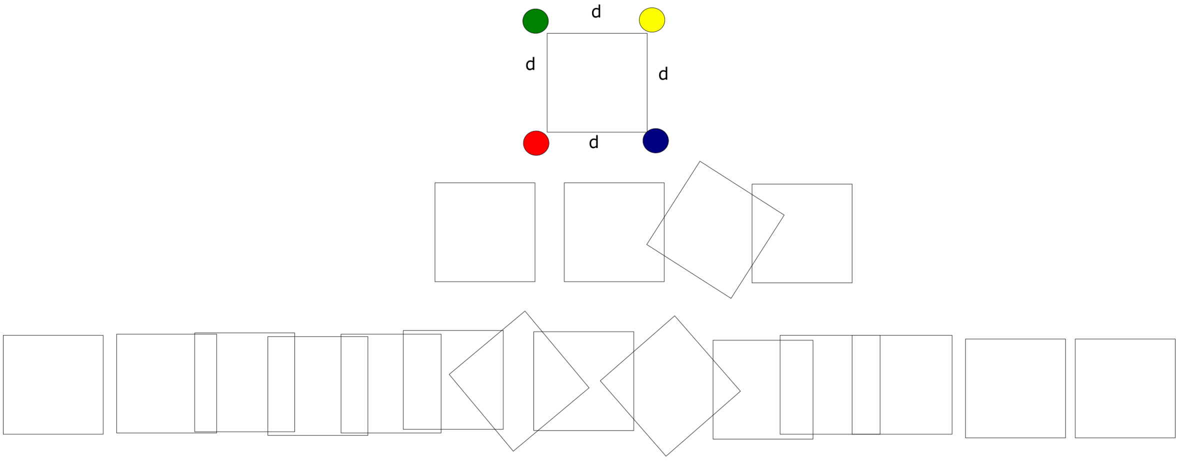


Figure 5.9: A figure representing a realistic intersecting graph D_{N^d}. The intersections are not representing an optimal peak, but rather happen in an uniform fashion.

5.7.8 Local Diversity of Models and Distribution of Paths in $G_{FullN_k^d}$

We already argued that under some assumptions, locally implemented diversity within each refinement set, improves the diversity of the explored solutions in comparison to not using diversity, but simply choosing optimal solutions.

In order to show that the graph $G_{FullN_k^d}$ is closer to the ideal graph compared to G_{FT-k}, just the dispersion of the nodes globally at level l is not sufficient. The second important aspect is the distribution of number of paths, which reach each node in the graph, we need that to be closer to uniform in $G_{FullN_k^d}$ with the help of diversity, compared to G_{FT-k}. The distribution of number of paths reaching each node at level l depends on the intersections of the neighborhoods of different refinement sets.

Generally speaking, we claim that using diversity will lead to fewer intersections between neighborhoods and the distribution of paths in $G_{FullN_k^d}$ will be closer to a uniform distribution compared to simply using optimality neighborhoods without diversity in graph G_{FT-k}. We will impose certain conditions with respect to the neighborhood intersections, which lead to rarer and more uniform intersections of paths and explain why these conditions are reasonable.

First, the intersections of the refinement sets of two models is proportional to how similar they are.

Condition 5.7.8 *Let $m_i, m_j \in \mathcal{M}$. Then $|r(m_i) \cap r(m_j)| \propto d(m_i, m_j)$.*

This can be transferred also to the diverse neighborhoods, defined over those refinement sets.

Condition 5.7.9 *Let $m_i, m_j \in \mathcal{M}$. Then $|N_d^k(r(m_i)) \cap N_d^k(r(m_j))| \propto d(m_i, m_j)$.*

Imposing Conditions 5.7.8 leads to less neighborhood intersections overall. This is due to the fact that maximal diversity as well as average pairwise diversity increases at each step in the Widening Nr_k^d.

As we stated, we assume that if two models are distant, their refinement sets and the diverse neighborhoods built from these refinement sets will be dissimilar. Then, we can conclude that, if we have three dissimilar models m_1,m_2 and m_3, they have dissimilar refinement sets dissimilar diverse neighborhoods, built from their refinement sets, $N_d^k(m_1)$, $N_d^k(m_2)$ and $N_d^k(m_3)$ are very dissimilar and have very few models in common, if any, which follows from Conditions 5.7.8,7.6.4. We also assume that if $S_{1,2}$, $S_{2,3}$, $S_{1,2}$ are their respective intersections $S_{1,2} = N_d^k(m_1) \cap N_d^k(m_2)$,$S_{2,3} = N_d^k(m_2) \cap N_d^k(m_3)$, $S_{1,3} = N_d^k(m_1) \cap N_d^k(m_3)$, then $S_{1,2}, S_{2,3}, S_{1,2}$ are also dissimilar from each other.

Furthermore, as demonstrated in Figure 5.7, due to the imposed diversity within the neighborhoods, the intersection of diverse neighborhoods are spread out instead of clustered together, as it would be if diversity within the neighborhoods is not used.

Based on these assumptions, we claim that the overall distribution of paths in the Widening graph $G_{FullN_k^d}$ will in the general case be closer to uniform compared to the distribution of paths reaching each node in $G_{FullTop-k}$, where diversity is not used. This is because the intersections of the diverse neighborhoods are less than the intersections of optimality neighborhoods and these intersections will be more uniformly distributed, compared to the intersections of paths in $G_{FullTop-k}$.

5.8 Search Space Partitioning and Widening via Neighborhoods

As it was already stated, Widening via neighborhoods can be presented as random walks on a particular neighborhood graph. Using diversity, we aim at building a graph $G_{N_k^d}$ which is as close to the ideal graph, described in 5.4.2, as possible. If the graph is built properly, each node of the Widening graph at level l will be on a different peak of the search space landscape. If, the graph is built properly, the random walks on such a graph can approximate search space partitioning, because each random walk will be exploring a peak in the search space. Additionally, exploitation via Widening via similarity neighborhoods can be added to the diverse Widening graph, in order to fine-tune the search in promising areas. Namely, using Widening via similarity neighborhoods, we can additionally explore the vicinity of the already discovered set of models with high diversity and high model quality and obtain further improved solutions. This again can approximate a search space partitioning, if different groups of parallel workers are assigned promising peaks in the landscape.

5.9 Conclusions.

In this chapter, we demonstrated the weaknesses and valuable properties of Widening using neighborhoods. Widening via optimality neighborhoods can emulate $Top - k$ Widening, without the necessity of communication, for a large enough number k of parallel workers. Widening via similarity neighborhoods can be used for exploitation of promising areas of the search space, and for a large enough number of parallel resources k, it can guarantee that the solutions discovered will be at most distance δ from the best solution in this interval.

With Widening via neighborhoods, we aim to build a graph, which explores all the important areas of the search space. By "important" we refer to different peaks in the landscape of the space of potential solutions, as described in 5.4.2. Although, Widening via diverse neighborhoods cannot guarantee that an ideal graph will be built in the general case, the graph generated by this method is closer to the desired one than that generated by simple Widening via optimality neighborhoods. Using local diversity by Widening via diverse neighborhoods leads to greater maximal and average diversity of the explored solution set Δ_{max} and Δ_{avg}, respectively. It is important to note, that each of the discussed graph structures of the neighborhood-based methods, contain the greedy path.

The flaws of Widening via optimality neighborhoods, just like those of $Top-k$, are related to converging to a local optimum. Widening via diverse neighborhoods tackles this problem and by introducing diversity locally via neighborhoods, leads to globally diverse exploration. Furthermore, due to the properties of diverse neighborhoods the intersections between different neighborhoods will happen more uniformly, the distribution of paths P^l will be more uniform in comparison to Widening via optimality neighborhoods. In order to be able to make stronger claims for the properties of the diverse Widening graph, we need to use *global diversity* approaches which do not only focus on each refinement set separately. A global approach to diversity will guarantee diverse behavior of the parallel workers in the search space without the necessity of communication.

We will discuss global diversity approaches in Chapter 6.

Chapter 6

Global Approaches: Partitioning of the Search Space Lattice for Refinement Operators of Type 1 and Widening via Global Diversity

6.1 Global Model-based Diversity and the Lattice Structure $L_{\mathcal{M}}$

In this chapter we will refer to a direct refinement of a model as a refinement. Global diversity approaches are based on defining individualized preferences for the parallel workers towards specific model fragments. These approaches can be model-based or data-based. Unlike the local approaches to diversity, where models are assigned diverse preferences. However, these assignments are valid only in a given refinement set. The global approaches assign diverse preferences, which apply to the whole search space, regardless of the step or the particular refinement set. For example, if parallel worker w_i will never select a set of model refinements M^{w_j} that will for sure be selected by w_j, the models discovered by these two workers are at least a distance $d = |M^{w_j}|$, where d is the Hamming distance with respect to model fragments $d(m_i, m_j) = |m_i \setminus m_j \cup m_j \setminus m_i|$. Global diversity does not suffer from the flaws of local approaches, where the parallel workers are assigned diverse models but only valid within a given refinement set. Within two different refinement sets, two different parallel workers can be assigned very similar, or the same model. The drawback of the global approach is rigidity. It is inflexible since the preferences of each parallel worker are assigned prior to the search irrespective of model performance. This can lead to the search not discovering good or optimal models.

6.2 Global Diversity

In contrast to the local approaches of Widening via neighborhoods, approaches based on global diversity are based on assigning preferences to parallel workers, which are valid for the whole search space, and not valid only within a given neighborhood. An example of global diverse approaches was already discussed in Chapter 3, where different preferences (model and data-based) were assigned to different parallel workers.

Proposition 6.2.1 *Let models m_i, m_j be two models in the lattice structure $L_{\mathcal{M}}$ and let $d = d(m_i, m_j)$ be the model-based distance between the two. Then the infimum of the models m_i and m_j, $inf(m_i, m_j)$ is at d refinement steps from the more general of the two models.*

This proposition helps us evaluate the potential intersection of the refinement sets of two models.

Corollary 6.2.2 *Let m_i, m_j be two models in the family of models $\mathcal{M}$. Then their refinement sets intersect nontrivially $|r(m_i) \cap r(m_j)| > 0$ iff $m_i, m_j \in r(m)$ for some model $m \in \mathcal{M}$.*

Proof Follows from Proposition 6.2.1 and the properties of the lattice.

It follows from Proposition 6.2.1 and Corollary 6.2.2 that if we know the distance d between two models m_i, m_j we can guarantee that within a known number of steps $n(d)$ the refinements $r(m_i), r(m_j)$ do not intersect. We will look only at the Hamming distance.

Proposition 6.2.3 *Let X be a set of model fragments, and let $S_i \subset X, 1 \leq i \leq k$, be pairwise disjoint subsets of order $|S_i| = d$. Suppose that for $1 \leq i \leq k$ each parallel worker w_i must look for the solutions, which contain the model fragments from S_i and do not contain model fragments from $\bigcup_{j \neq i} S_j$. Then at level $l \geq d$, the set of the corresponding solutions $\{m_1, \ldots, m_i\}$ satisfies $d(m_i, m_j) \geq 2d, 1 \leq i < j \leq k$.*

In order to use *both* diversity and performance, the above results should be combined with the model performance. That is, for this type of diverse Widening to be successful, it is important to look for the best models, which satisfy the diversity requirements. The smaller the imposed minimal distance d, the closer the search of each parallel worker to the greedy search.

6.3 Drawbacks and Benefits of the Global Diversity Approaches.

The biggest drawback of the global diversity approaches is their lack of flexibility when assigning the different forbidden/allowed model fragments because this assignment is done without consideration of the model performance. Well-performing models can be ignored due to rigid assigning of these preferences. However, given enough k and small d the density of exploration of the search space at level l is sufficient to discover the good models.

6.4 Tradeoff Between Diversity and Model Quality

Diversity requirements can be and should be, combined with selecting models of the best possible performance. However, the stronger the diversity requirements, the more rigid this approach, and the more difficult it is to choose models with good performance. Widening via global diversity can be combined with requirements for model quality by using different weight parameters for diversity and optimality:

$$\alpha * f_c(m) + \beta * \psi(m),$$

where $f_c(m)$ is a function, which evaluates how well the model satisfies the imposed diversity requirements.

6.5 Symmetric Chain Decomposition and Widening

Widening, where the full search space is explored, is related to the problem of full enumeration. For lattices, many approaches of full enumeration exist. We will describe one approach, the properties of which make it very useful for Widening. First, we will introduce important definitions.

Definition Given a family of models $\mathcal{M}$, and a refinement operator r, which introduces a partial order on $\mathcal{M}$. We will refer to as *antichain of size n* the set of models $m_1, \ldots, m_n$, where any two distinct elements are not comparable with each other. Namely, no model is a refinement of another model from the set $(m_i \notin r(m_j)), \forall i \neq j, i, j \in \{1, \ldots, n\}$. Adapted from [58].

The following definition is common in order theory, but we adapt it to suit the context of Widening.

Definition Given a refinement operator r and a family of models $\mathcal{M}$, we will call the ordered set of models $m_1, \ldots, m_n$, where $m_{i+1} \in r(m_i)$, $1 \geq i \leq n-1$ a *chain* of models. Adapted from [58].

Definition Given a refinement operator r and a family of models $\mathcal{M}$, we will call the ordered set of models $m_1, \ldots, m_k$, where $m_{i+1} \in r(m_i)$, $1 \geq i \leq k-1$ a *symmetric chain* of models if $|m_1| + |m_k| = n$ and Adapted from [74].

Definition Given a model family $\mathcal{M}$, and a refinement operator r, we call an antichain of models $m_1, \ldots, m_n$ a *maximal antichain*, if it is an antichain and it is at least as large as every other antichain. The *width* of a partially ordered set is the cardinality of a maximum antichain. Adapted from [58].

Observation 6.5.1 *In Ideal Widening, the final solution set of models should be an* antichain *of models.*

A very important result in lattice theory is Dilworth's theorem for decomposition of a poset. We will use it for a lattice.

The theorem is quoted directly from [58].

Theorem 6.5.2 ***Dilworth's theorem:***
Any finite poset $P = (X, \leq)$ of width w can be decomposed into w chains. Furthermore, w chains are necessary for any such decomposition. The width *of a poset is equivalent to the minimal number of the chains, necessary to decompose the poset.*

The result of this theorem and multiple algorithms, associated with it, can be used in Widening. The theorem is quoted directly from [74].

Theorem 6.5.3 ***Sperner's theorem:***
The width of a Boolean lattice $\mathcal{B}_n$ is $w = \binom{n}{\lceil \frac{n}{2} \rceil}$

From this and the Dilworth's theorem, it follows that the maximal number of parallel workers needed for full lattice exploration is $\binom{n}{\lceil \frac{n}{2} \rceil}$, where n is the number of model fragments because this is the width of the lattice, the refinement level with the greatest number of refinements.

The definition is adapted from [58].

Definition *Chain decomposition* is an approach, in which the lattice is decomposed into non-intersecting chains of models.

It is a known fact, that Boolean lattices have a special type of chain decomposition, *symmetric chain decomposition (SCD)*,[75]. This fact can also be used for Widening, where different chains can be assigned to different parallel workers. Symmetric chain decomposition is, in fact, a partitioning of the lattice into non-intersecting chains.

Below, we present the *Greene-Kleitman rule* [75] to grow a chain, adapted from [90]. It is based on the bit-vector encoding of the lattice, where each model is presented as a binary vector of length n. The set of model fragments is ordered and each bit in the bit vector encoding of the model corresponds to a model fragment, based on that order. For each model, the value for a bit is 1 if the corresponding model fragment belongs to the model and 0 if the model fragment does not belong to the model. View 1 bits as right brackets and 0 bits as left brackets and in each vector, match the brackets left with right following the usual rules for matching brackets. For example, in $0_1 0_2 1_2 1_1 0_3 0_4$ the first and second bit are left brackets and are matched with the fourth and third bit, respectively, which are right brackets, or (()))) . A chain is grown by starting with a vector with no unmatched 1. The first unmatched 0 is changed to 1 in order to get its successor. Continue until a vector with no unmatched 0 is reached. For example,

$$0001 \rightarrow 1001 \rightarrow 1101.$$

An example of a symmetric chain decomposition of a powerset lattice is shown in Figure 6.1. Greene and Kleitman showed in [75] that this rule gives a symmetric chain decomposition of the Boolean lattice. The number of chains from such a decomposition is always $\binom{n}{\lceil \frac{n}{2} \rceil}$.

We know that each chain is uniquely determined by its first element, (we will call it *head of the chain*). The remaining problem is assigning the first element of each chain to a unique parallel worker. This can be computationally intensive, which limits the applicability of the approach.

Once each parallel worker reaches the head of the chain, it can use the chain generating a rule to grow its chain, without the risk of intersection with another worker's path. Each model is reachable along these chains, so this is, in fact, the ideal partitioning of the search space, that was defined as *path-closed partition Widening* in Chapter3. The only issue is the load balancing because different chains are of different size. However, if the number of chains is much larger than the number of parallel workers, and the chains are assigned randomly to the parallel workers, then the workloads will be more balanced.

Observation 6.5.4 *If each chain, obtained via SCD of the search space lattice is assigned to a unique parallel worker, this partitioning fulfills the criteria of path closed Widening, as described in Chapter 3, Definition 3.1.1. Each chain is path-closed, and, moreover, there is no intersection between the chains.*

In fact, we can use the symmetric chain decomposition, to define a set of perfect selection operators, but only if the search could start at the head of each chain. This however, would require to generate each head and is very computationally intensive.

Note 6.5.5 *We know how to recognize if a given model is a head of a chain – a head of the chain is such an element, which has no unmatched 1's,[90].*

We need to assign to each parallel worker a chain C_i (or more chains) and a path from the empty model m_0 to the head of the chain. Instead of generating the *heads* of the chains, we want for each chain to be reachable by a parallel worker from the empty model. We want that each chain is explored by a unique parallel worker, but that to each chain we add a subpath from the empty model to the head of the chain. Namely, at the beginning of the search, the search paths will have intersections until each parallel worker reaches the head element of its chain. This will implement, what was defined as **approximate partition-Based Widening** in Chapter 3, where there is an intersection at the beginning of the search paths, but then as the search progresses, there is no intersection. If the parallel workers are less than the number of chains, we can assign several chains per parallel worker. To generate the necessary subpaths leading to the chains' heads, we will use the following observation.

Lemma 6.5.6 *Each first element of a chain (a head) can be reached only from another first element of a chain. Namely, each first element of a chain is a direct refinement only to other heads of chains.*

Proof Follows from the fact that a model can be a head only if it has no unmatched 1's. The direct refinement operation flips 0 to a 1 in some position in the bit vector So if an element already has an unmatched 1, by having another 1 instead of an unmatched 0, the model cannot become a head of a chain, it will still have unmatched 1s.

We already know from Chapter5, that partitioning a direct refinement set among parallel workers is easy since each worker can be assigned apriori a different model from a given refinement set. What is problematic is assigning different models, when there are several refinement sets on a given level l, with common elements, as is the case in a lattice. Each model on level l is reachable as a direct refinement from many models at level $l-1$. We need to forbid the access to the heads at level l from more than one head at level $l-1$. We can accomplish that using a traversal order on the heads from level l to be reached from the heads of level $l-1$ as follows: The heads that can be reached by head 1 from level $l-1$ will not be allowed to be reached from any other head, the heads at level l that can be reached from head 2 of level $l-1$, but not from head 1 will not be allowed to be reached from any other head, but head 2, and so on.

Lemma 6.5.7 *Each head at level l must have at least l 0 bits before the position of its last 1 bit. Its last 1 bit can have a position at least* $2l$ *or greater.*

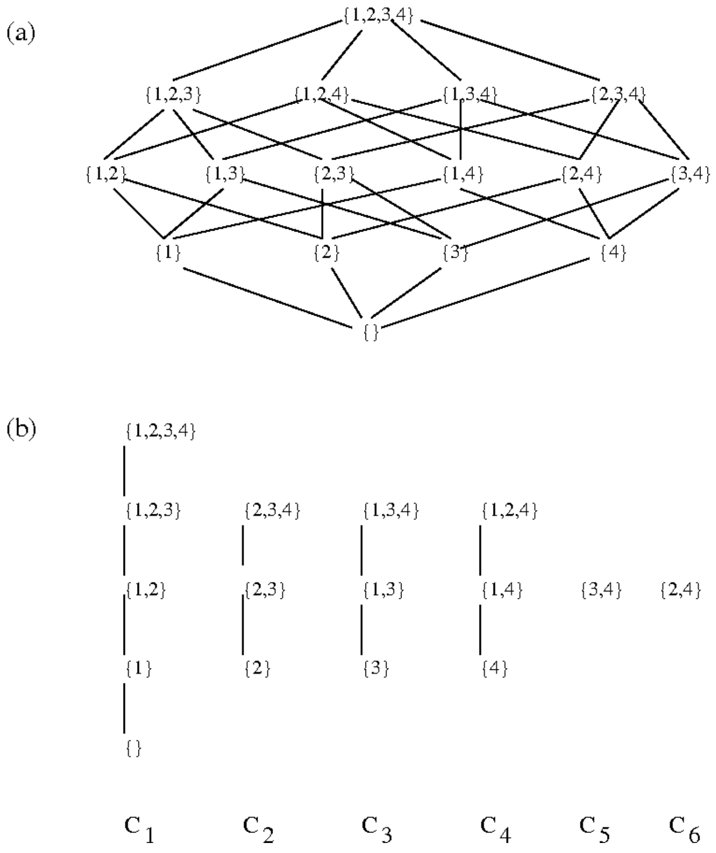


Figure 6.1: A symmetric chain decomposition of a Boolean lattice of $n = 4$ by the Greene and Kleitman method. The figure is taken from [99].

Proof At level l, each model has l 1's in its bit vector. If it has less than l 0's before its last 1 bit, then it has an unmatched 1 and cannot be a head of a chain.

Definition Given a bit vector representation of a model m of length n, and let the positions of the bits be numbered $0, \ldots, n$ we will call the last 1 bit in the bit vector, the bit with the greatest position number.

Lemma 6.5.8 *If a head model at level $l - 1$ has its last 1 bit on position $p \leq 2l - 1$ then its reachable heads are generated by 0-to-1 bit flips at one from the bit positions $2l, 2l + 1, \ldots, n$. Each bit flip generates a new head at level l. If $p > 2l - 1$, then the 0-to-1 bit flips, which are used to reach the heads at level l are at positions $p, p+1, \ldots, l$.*

Proof We will show that in this way all elements with no unmatched 1s at level l are generated from heads at level $l-1$. A head at level $l-1$ has $l-1$ 1s, all matched. Then, it has at least $l-1$ 0s. To have no unmatched 1s at level l there must be at least l 0s in the first $p+2$ positions, where p is the position of the last 1 bit. If $p > 2l-1$ there is at least one unmatched 0 before p and flipping any bit after p will have a 0 bit to match. If $p < 2l-1$, then there will be an unmatched 0between p and $2l$, which can be its match. If $p = 2l-1 > 2l-2$ there is an unmatched 0 before p, which can be the match if the $2l$ bit is flipped to 1.

Lemma 6.5.9 *Flipping a first non-matching 0 to 1 in a head model, cannot create a head.*

Proof The first non-matching 0 bit is either the first bit of the head, or directly behind a 1 bit. If we shift such a non-matched 0 bit, which is a first bit, to 1, there is no 0 bit in front of it to match it. If we flip a first non-matching 0, directly behind a 1 bit, it will need to be matched by a 0 bit before the matching 0 of the 1 bit in front it. But since it is a first matching bit, there is no such unmatched 0 in the front of it.

Theorem 6.5.10 *Let models $m_1, \ldots, m_m$ be all the heads at level $l-1$, and let $p_1, \ldots, p_m$ be the positions of their last 1 bit. Let $max_i = \max(2l, p_i + 1), i = 1, \ldots, m$. Then if we generate the $n - max_i$ direct refinements by flipping each 0 bit in positions $max_i, max_i + 1, \ldots, n$ for each head m_i, $i = 1, \ldots, m$ we will generate each possible head at level l and generate it exactly once.*

Proof First, let us assume that a head at level l, m_h, is not generated by the described procedure. We know that m_h is a direct refinement of some model m_h^{l-1} in $l-1$, and that m_h^{l-1} is also a head (follows from lemma 6.5.6). Then, if the model m_h^l is not reached by the procedure above, it means that it was reached by flipping a 0 bit with a position $p < max, max = \max(2l, p_h + 1)$, where p_h is the position of the last 1 bit in m_h^{l-1}. If $max = 2l$, then m_h^l has less than l 0 bits before its last positioned 1 bit, as it has l 1 bits and the position of its last 1 bit is less than $2l$. From lemma 6.5.7, m_h^l cannot be a head. Let $max = p_h + 1$, then the head m_h^l is reachable by a flip of a non-matching 0, with position $p < p_h$. But then let us consider the model m_r^{l-1}, which has a 0 bit at position p_h and 1 bit at position p. The model m_r^{l-1} is a head, because the 1 bit at position p is matched, and there is no other changes in the other 1 bits, they are still matched, so it has $l-1$ matched 1 bits, which means it is a head at level $l-1$. But we know that the last 1 bit in m_r^{l-1}, positioned at the p_r position, $p_r < p_h$. Then model m_h can be reached by m_r^{l-1}, which follows the described procedure and thus is a contradiction with the assumption.

Now we will show that via the described procedure, each head at level l is reached no more than once from the heads at level $l-1$.

Let us assume that m_h was reached by two heads from level $l-1$, following the described procedure, m_h^{l-1} and m_r^{l-1}. Let us assume m_h was reached from m_h^{l-1} by a flip of the bit in position p_1 and from m_r^{l-1} by a flip in position p_2, w.l.g let $p_1 < p_2$. Let the positions of last 1 bits in m_h^{l-1} and m_r^{l-1} be p_h and p_r respectively. However, since both of them reach m_h by a different bit flip, p_1 must be 1 for m_r^{l-1} and p_2 must be 1 in m_h^{l-1}. Since $p_2 > p_1$, then in to reach m_h, m_h^{l-1} a bit was flipped, which was before the last 1 bit. So the procedure was not followed, which is a contradiction.

We will illustrate the procedure described above by a simple example.

Example Consider the Boolean lattice, for $n = 5$. In 01000, the last 1 is at position 2, so we can flip the non-first unmatched 0s, after the last 1 bit, so positions 4 and 5. Reaching all the heads at level $l = 3$ from the models from $l = 2$ can be done as follows: $01000 \to 01010, 01000 \to 01001, 00100 \to 00110, 00100 \to 00101, 00010 \to 00011$.

We will outline the following simple observations, which follow directly from the Theorem 6.5.10.

Observation 6.5.11 *If a head model at level $l-1$ has its last 1 bit on position $p \leq 2l-1$ then all its reachable heads are generated by 0-to-1 bit flips at one from the bit positions $2l, 2l+1, \dots, n$. Each bit flip generates a new head at level l. If $p > 2l-1$, then the 0-to-1 bit flips, which are used to reach the heads at level l are at positions $p, p+1, \dots, l$.*

Let the subpaths to the head elements be represented as a sequence of bits to be flipped.

Observation 6.5.12 *If $(x_1, \dots, x_q)$ is a path to a head, built, following the procedure described in Theorem 6.5.10, then $(x_1, \dots, x_{q-1}), (x_1, \dots, x_{q-2}), \dots, (x_1)$ are also paths to heads.*

Observation 6.5.13 *If $(x_1, \dots, x_q)$ is a path to a head at levelq, built, following the procedure described in 6.5.10 and if $x_q > 2q$ then $(x_1, \dots, x_q, x_{q+1} = x_q+1), (x_1, \dots, x_q, x_{q+1} = x_q+2), \dots, (x_1, \dots, x_q, x_{q+1} = x_q+n-q)$ are also paths to heads. Else, if $x_q = 2q$ then $(x_1, \dots, x_q, x_{q+1} = x_q+2), (x_1, \dots, x_q, x_{q+1} = x_q+3), \dots, (x_1, \dots, x_q, x_{q+1} = x_q+n-q-2)$ are also paths to heads.*

These observations help us generate the paths to the heads of the chains from the empty model.

Example Let $n = 6$ At level 1 the head 000000 has no 1s, so each non-first non-matching 0, when flipped generates a head: $010000, 001000, 000100, 000010, 000001$. The subpaths are $(2), (3), (4), (5), (6)$. At level 2, from (2) the bits that can be flipped are at the positions $4, \dots, 6$, for $\{3\}$ also $4, \dots, 6$, for (4), $5, 6$, and for (5) only the bit at position 6 can be flipped. So we have the following paths: $(2,4)$,$(2,5)$,$(2,6)$,$(3,4)$,$(3,5)$,$(3,6)$, $(4,5)$,$(4,6)$,$(5,6)$. For $l = 3$, recursively, for the already existing paths $(2,4)$, $(2,5)$, $(3,4)(3,5)$, $(4,5)$ only the 0 in the position 6 can be flipped, so the paths are $(2,4,6)$,$(2,5,6)$,$(3,4,6)$,$(3,5,6)$,$(4,5,6)$.

We can use these observations to build the subpaths from the empty model to each head, but for a large n it is very computationally expensive, due to the large number of chains. We need an efficient way to generate these subpaths.

6.6 Conclusions

We know that for a particular type of families of models, the search space has a lattice structure. We use this property to partition the whole search space among a fixed number of parallel workers, as well as apply Widening approaches that guarantee model-based diversity of the explored models. Global diversity approaches can provide global diversity guarantees, unlike local approaches, discussed in the previous chapter. However, the drawbacks are the rigidity of this approach, which follows the fixed assignment of forbidden/allowed sets of model fragments. These assignments are done apriori and do not take into account the performance of the models. This may lead to exploring models that are not of good quality, especially when larger diversity is enforced. Global approaches can also be data-based, but in that case, the lattice structure cannot be applied.

Chapter 7

Widening of the Greedy Algorithm for the Set Cover Problem

This chapter is adapted from [86, 7, 88].

7.1 The Set Cover Problem

In this chapter, we will use the set cover problem (SCP) to illustrate the benefits of different Widening approaches. The set cover problem underlies quite a few data mining algorithms, for instance when trying to find the smallest number of itemsets or rules, which explain the data, finding minimal explanations for patterns, in classification, data quality assessment, and in information retrieval, [29]. We apply the Widening approaches to the greedy algorithm for SCP. We have already this algorithm in [7], [86], [88],[89]. to illustrate the benefits of Widening.

7.2 Formalization of the Set Cover Problem

Because finding an optimal solution for the SCP is NP-hard [98], a heuristic approach is preferred: a greedy algorithm, which at each step selects the subset with the largest number of uncovered elements, is widely used. This efficient and simple greedy algorithm was shown to perform perform surprisingly well in [91]. Namely, it guarantees an approximation ratio of $H(n)$, where $H(n)$ is the n-th harmonic number and n is the number of elements, which need to be covered.

We consider the standard (unweighted) set cover problem. Given a set X of n elements and a collection $\mathcal{S}$ of m subsets of $X : \mathcal{S} = \{S_1, S_2, \dots, S_m\}$. We assume that the union of all of the sets in $\mathcal{S}$ is X, with $|X| = n$: $\bigcup_{S_i \in \mathcal{S}} S_i = X$. The aim is to find a sub-collection of sets in $\mathcal{S}$, of minimum size, that covers all elements of X.

7.2.1 Greedy Set Covering

The greedy algorithm [91] attempts to construct the minimal set cover in the following way. It starts with the empty set being the temporary cover and at each step selects and adds a single subset to it. The subset selected is the one which contains the most elements that are not yet covered by the temporary cover. To be consistent with the terminology previously defined: if C is the temporary cover, a refinement generated by $r(C)$ represents the addition of a single subset, not yet part of C, to C. From all the possible refinements, generated by $r(\cdot)$, the one with the largest number of elements is chosen as the new temporary cover ($s(\cdot)$ function). Algorithm 1 illustrates this procedure.

Algorithm 1: Greedy Algorithm for Set Cover Problem,

```
Data: collection $\mathcal{S}$ of sets over universe $X$
Result: set cover $C$: $\bigcup_{S \in C} S = X$
$C \leftarrow \emptyset$;
repeat
    $S_{\text{current}} = \bigcup_{S \in C} S$
    $S_{\text{best}} = \arg\max_{S \in \mathcal{S}} \{|S \backslash S \cap S_{\text{current}}\}$
    $C \leftarrow C \cup S_{\text{best}}$
until $\bigcup_{S \in C} S = X$;
return $C$.
```

7.3 Widened Set Covering.

7.3.1 Top-k Set Cover.

In contrast to the greedy algorithm, the Widening of the greedy algorithm builds k temporary covers in parallel. The focus in this algorithm is to use resources to explore a large number of refinements in parallel. The number of parallel workers k is referred to as the Widening parameter, or width. The choice of a value for the parameter k depends on the available compute resources.

A single iteration of the widened algorithm then operates as follows. Let $C_1, \cdots, C_k$ represent the k temporary covers. A refinement of C_i is created by adding a new subset to C_i. For each C_i, the k refinements which contain the largest number of elements, are selected. This results in k^2 refinements in total. From those, the top k refinements are selected, resulting in k new temporary covers $C_1^{'}, \cdots, C_k^{'}$. As we will see later, the quality of the solutions will increase with larger k, due to more options being explored in parallel.

Algorithm 2 shows the pseudo-code of the widened set covering. The standard iterative algorithm [91] follows a greedy strategy, which, at each step, selects the subset

Algorithm 2: Top-k Widening of the greedy algorithm for SCP.

```
Data: collection $\mathcal{S}$ of sets over universe $X$, number of parallel resources $k$
Result: set cover $C$: $\bigcup_{S \in C} S = X$
$C_1 \leftarrow \emptyset, C_2 \leftarrow \emptyset, \ldots, C_k \leftarrow \emptyset$;
repeat
    foreach $C_i, i \in \{1, \ldots, k\}$ in parallel do
        /* $r(\cdot)$                                                    */
        $S_{\text{current}_\text{i}} = \bigcup_{S \in C_i} S$
        foreach $j \in \{1, \ldots, k\}$ do
            $S_{\text{best}_\text{i,j}} = \arg\max_{S \in \mathcal{S}} \{|S \backslash S \cap S_{\text{current}_\text{i}}| \ : S \in \mathcal{S} \setminus \{S_{i,1}, \ldots, S_{i,j-1}\}\}$
            $C_{i,j} \leftarrow C_i \cup S_{\text{best}_\text{i,j}}$
    /* $s(\cdot)$                                                        */
    $C_1 \leftarrow \max\{C_{i,j}\} \ : \ i \in \{1, \ldots, k\}, j \in \{1, \ldots, k\}$;
    ...
    $C_k \leftarrow \max\{C_{i,j} \backslash\{C_1, \ldots, C_{k-1}\}\}$;
until $\exists i, i \in \{1, \ldots, k\} : \bigcup_{S \in C_i} S = X$;
return $\min\{C_i\}, i \in \{1, \ldots, k\}$.
```

with the largest number of remaining uncovered elements. Using the formalizations introduced above, a single iterative step of the algorithm operates as follows: if C is the temporary cover, a refinement generated by $r_{\text{greedySCP}}(m)$ represents the addition of a single subset, not yet part of m, to m. From all of the possible refinements, generated by $r_{\text{greedySCP}}(m)$, $s_{\text{greedySCP}}$ picks the one with the largest number of covered elements as the new intermediate cover. The quality measure ψ, used by the selection operator, $s_{greedySCP}$, therefore simply ranks the models based on the number of elements they cover.

7.3.2 Diverse $Top - k$ Widening

Instead of selecting one locally best intermediate cover, the $Top - k$ Widening of the greedy SCP algorithm selects k best covers at each given step. To implement diversity, we can use a simple threshold based on the Jaccard distance and enforce that the chosen k intermediate covers chosen by the selection operator $s_{Top-k,\delta}$ at each step have a minimum distance:

$$d(m_i, m_j) = 1 - \frac{|m_i \cap m_j|}{|m_i \cup m_j|}.$$

(Each model m covers a set of elements, so we are interested in picking intermediate models that are sufficiently different.)

7.3.3 Communication-less Widening.

Global Approach: Model-based Diversity.

Enforcing diversity without continuously comparing intermediate models is more difficult. We can define individual quality measures ψ_i, by enforcing different preferences for different subsets. Given an intermediate cover m, ψ_i evaluates the refinement $m' = m \cup S_j$ for an additional subset S_j based on the original quality measure and an individual preference weight $w_i \in (0, 1)$ for the subset S_j:

$$\psi_i(m \cup S_j) = \psi(m \cup S_j) + t * w_i(S_j).$$

The set of weights $w_i(\cdot)$ for a given ψ_i defines an order π_i on the set of subsets $\mathcal{S}$ for a particular parallel worker i:

$$w_i(S_{\pi_i(1)}) > \cdots > w_i(S_{\pi_i(|\mathcal{S}|)}).$$

Our goal is to have k diverse orders $\pi_1, \ldots, \pi_k$ of the subsets by ensuring that the *inversion distances* between different orders are large. The inversion distance between two ordered sets calculates how many pairs of elements are present in a different order in the two orders π_p and π_q:

$$d_{\text{inv}}(\pi_p, \pi_q) = \sum_{k \neq l} \begin{cases} 1 & \text{if } (\pi_p(k) - \pi_p(l)) \cdot (\pi_q(k) - \pi_q(l)) < 0 \\ 0 & \text{else} \end{cases}.$$

Assigning preferences in this fashion will steer the selection operators based on characteristics of the models (or model fragments).

Global Approach: Data-driven Diversity.

In contrast to the model-driven diversity described above, we can also ensure diversity by weighting data elements. To accomplish this we enforce diverse preferences for the elements from X for the different selection operators s_i:

$$\psi_i(m \cup S_j) = \psi(m \cup S_j) + t \cdot \frac{1}{|\{e \in S_j \wedge e \notin m\}|} \sum_{e \in S_j \wedge e \notin m} w_i(e),$$

the preference for different elements is again defined via weights $w_i(e)$ and the weights define an ordering on the elements where we again aim for k different orderings via sufficient inversion distance. Note that this approach bears some similarities to boosting because we weight the impact of data elements on the model quality measure differently. It must be noted that, while using diverse quality measures can help steer the parallel workers into diverse selection paths, it by no means guarantees it. Choosing different models at each step can still lead to having the same final solution, just generated along a different path. In order to have guarantees, instead of different preferences in different orders for given model fragments or data points, partitioning of model fragments is needed, as described in Chapter 6. In the following sections, however, we will demonstrate that regardless of the lack of theoretical guarantees, this simple approach to diversity-driven Widening is beneficial.

7.4 Local Communication-less Approach, Widening via Neighborhoods.

In this section, we will discuss Widening approaches via neighborhoods, as described in Chapters 4, 5. Each neighborhood is built on the refinement set $r(m)$ of a given model m. Let $m = \{S_i\}, i = 1, \dots, l-1$. A refinement set $r(m)$ consists of a set of models $\{\{S_i\} \cup S_{j_1}, \{S_i\} \cup S_{j_2}, \dots, \{S_i\} \cup S_{j_{n-l+1}}\}, \quad i = 1, \dots, l-1, \quad S_{j_1}, \dots, S_{j_{n-l+1}} \notin \{S_i\}$, which differ in only one subset from each other, i.e. each of them contains $m = \{S_i\}, i = 1, \dots, l-1$ and exactly one additional subset.

Then a k-neighborhood within the refinement set will contain k sets of subsets, chosen from the refinement set of model m, which are chosen differently depending on the type of neighborhood. Each parallel worker is assigned one model from the k-neighborhood, with or without repetition. Below we will describe the different types of neighborhoods in the context of the SCP.

7.4.1 Widening via Optimality Neighborhoods

Given a model $m = \{S_i\}, |\{S_i\}| = l-1, i = 1, \dots, l-1$ the optimality k-neighborhood of $r(m) = \{\{S_i\} \cup S_{j_1}, \{S_i\} \cup S_{j_2}, \dots, \{S_i\} \cup S_{j_{n-l+1}}\}, \quad i = 1, \dots, l, \quad S_{j_1}, \dots, S_{j_{n-l+1}} \notin \{S_i\}$ consists of the best k models with respect to performance in $r(m)$.

7.4.2 Widening via Similarity Neighborhoods

Let $m = \{S_i\}, |\{S_i\}| = l-1, i = 1, \dots, l-1$, be a temporary solution. Let $M^r = r(m)$ be the refinement set of m, with m' being the optimal model in the set. Let metric d be an appropriate distance measure. Then the similarity k-neighborhood will consist of m' and the $k-1$ most similar models to m' within M^r. The only difference between each model in this refinement set is one subset. Therefore, this is equivalent to choosing the subset that contains the greatest number of uncovered elements and $k-1$ subsets which are most similar to it according to a metric d.

Issues with Preprocessing in the Context of SCP

Appropriate preprocessing can be based on methods which make use of model (model fragments) similarity $k-d$ trees and *local sensitivity hashing.* Using these approaches, we can assign to each model fragment its $k-1$ neighbors before the search starts. This is beneficial for the running time of the algorithm (especially if many runs are performed on the same data set). Each of the k parallel workers selects the neighbor assigned to it apriori, from a given neighborhood of the locally optimal model fragment. Note, that this similarity will be based on initial elements, and not based on the elements, which are still not covered at a step l. Preprocessing is especially important in Widening via diverse neighborhoods, where building the neighborhood dominates the calculation, and the running time is highly dependent on the size of the neighborhood, $O(n^2)$. A diverse neighborhood is a subset of the refinement set for model m that puts restrictions on Δ with the goal to improve search space exploration. As we discussed in Chapter 5, the goal is building a graph structure with desired properties, so that each parallel worker explores diverse and promising paths. There are different ways to introduce diversity within the neighborhood structures. We will give examples using fitness sharing and a simple threshold.

7.4.3 Widening via Diverse Neighborhoods

Diversity via Fitness Sharing

The main idea behind fitness sharing was discussed in Chapter 4. Here we will discuss how fitness sharing is used specifically for building diverse neighborhoods for Widening of the greedy heuristic for the SCP. In this case, each neighborhood consists of set covers that will differ from each other by a single subset. These k subsets are chosen not using the number of uncovered elements in them, as it is in the typical optimality neighborhoods, but also in their respective "rarity". The idea is to select representatives of different "subpopulations". In this case, fitness sharing enforces the selection of subsets from different groups of similar subsets. The fitness sharing does not use any preprocessing, as the other approaches inspired by the genetic algorithms. In the case of SCP, models who share fitness, are temporary covers, which are similar. These covers are members of the same refinement set, the difference between them is one subset. Niching is based on the similarity of single subsets.

Diversity via Threshold

In this approach, the diverse neighborhood is built using a threshold δ. At each step, given a model m, the diverse neighborhood $N_k^\delta(m)$ consists of the k most optimal subsets that are distance δ from each other, based on some distance measure. In this context the distance measures Jaccard or Hamming are appropriate. The runtime of this approach without preprocessing is in $O(n^2)$.

7.5 Methods.

All the approaches were implemented using KNIME [15]. All the experiments are performed on three data sets with different properties from the OR Library database. Each experiment was run 50 times with shuffled order of the data.

7.5.1 $Top - k$ Widening.

We compare the effect of the size of Widening on the quality of the obtained results. We use this Widening method with communication as a benchmark for comparison with our communication-less methods.

7.5.2 Widening via Local Neighborhoods.

Widening via Optimality Neighborhoods

We use Widening via optimality neighborhoods to investigate the effects of the parameters k and θ. We compared the quality of results using Widening via optimality neighborhoods for different parameters k with fixed θ as well as the quality of results as θ increases. Additionally, we compare the quality of results of Widening via optimality neighborhoods and Widening with communication, $Top - k$, in order to see whether the approaches with communication can compete to those without.

Widening via Similarity Neighborhoods

In order to demonstrate experimentally how Widening via similarity neighborhoods can be used for exploitation and similarity search, we use a very small neighborhood size θ with a large number of parallel resources. For exploitation, we are interested in the best performance of Widening via similarity neighborhoods, how it compares to the greedy algorithm and how it varies with the different neighborhood sizes and for different k. In order to demonstrate the potential suitability of Widening via similarity neighborhoods as similarity search, we evaluate both, the similarity of the obtained set of k models and the average performance of this set of models. The goal is to obtain similar models, which perform well. We compare the average similarity of the set of k resulting models to the average similarity of the k obtained models by Widening via optimality neighborhoods.

Widening via Diverse Neighborhoods.

We have chosen two approaches to building diverse neighborhoods, one is based on a simple threshold and the other uses the fitness sharing approach in order to build diverse neighborhoods. We are interested in whether the use of diversity improves the quality of the obtained result in comparison to Widening via simple optimality neighborhoods, without diversity. In addition, we investigate how diversity without communication compares to diverse $Top - k$, which uses communication. The main drawback of fitness sharing is that the computation of the shared fitness for the entire population in each generation can be very time-consuming. In [116], the niches for only a small subset of the population that is randomly sampled from the whole population are calculated. This dramatically improves the running time. We implemented this approach to fitness sharing. Instead of randomly sampling, another approach would be to remove the extremely poor performers.

7.6 Experimental Results

In this section, we will present and compare the experimental results of the different Widening methods applied to the greedy algorithm of the SCP.

Remark All the plots are made using R, [122]. The box plots visualize the performance of the Widening approaches. The black horizontal line is the median value. The bottom and top of the box are the first and third quartiles. For the ends of the whiskers the default positions, as defined in *boxplot.stats grDevices* from the R documentation, and are located at roughly 5% and 95% of the confidence interval. Any data not included between the whiskers is plotted as an outlier with a small circle.

7.6.1 $Top - k$ and Diverse $Top - k$.

By increasing the parameter k, the quality of discovered solution improves. The optimal, or beneficial values of diversity parameters are data dependent, and, if the value is selected properly, this leads to improvement of the quality of the obtained solution. Strongly diverse exploration for a small number of parallel workers leads to worse results due to randomization. Diversity combined with a sufficiently large number of parallel resources leads to improvement of the quality of solution found.

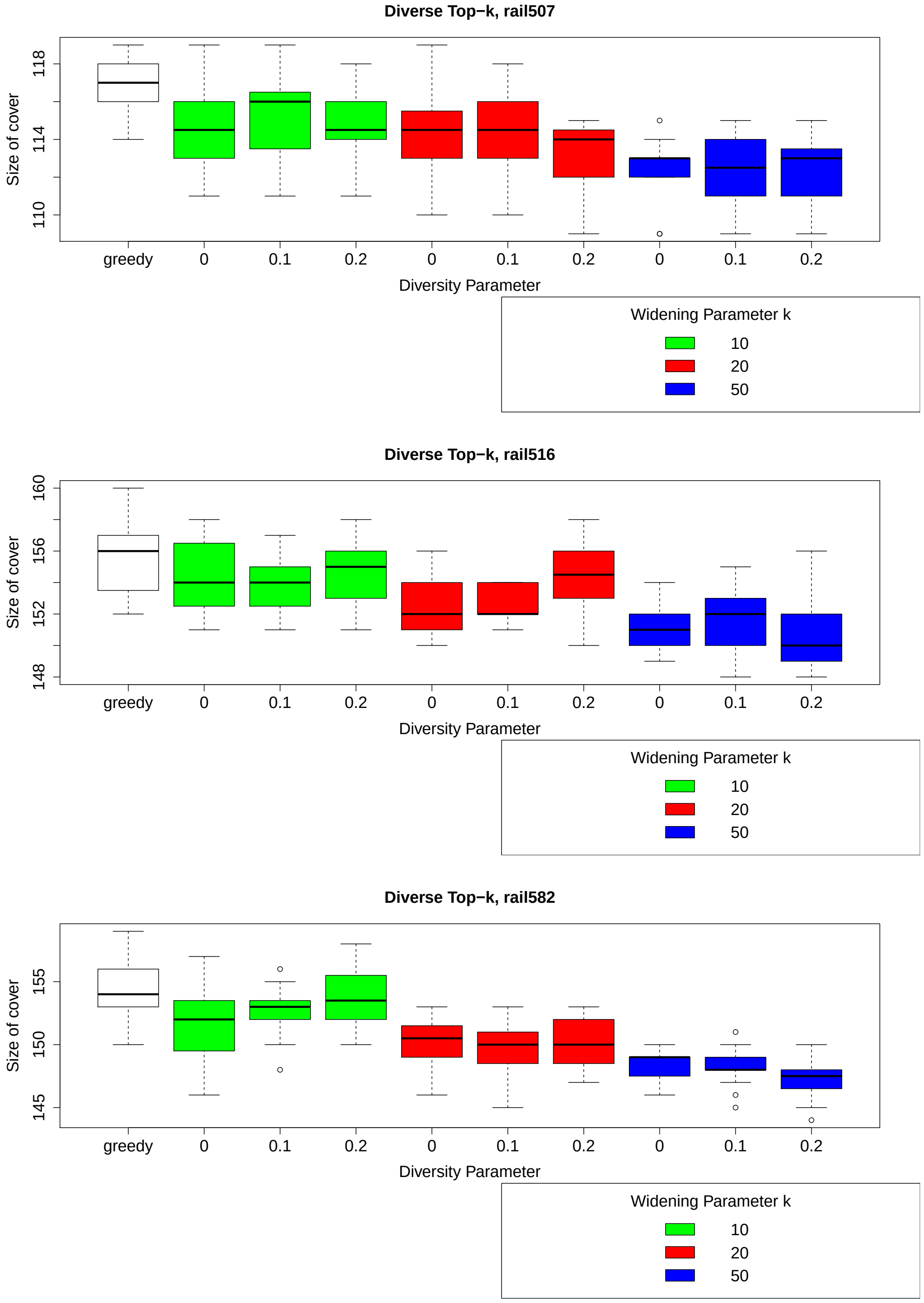
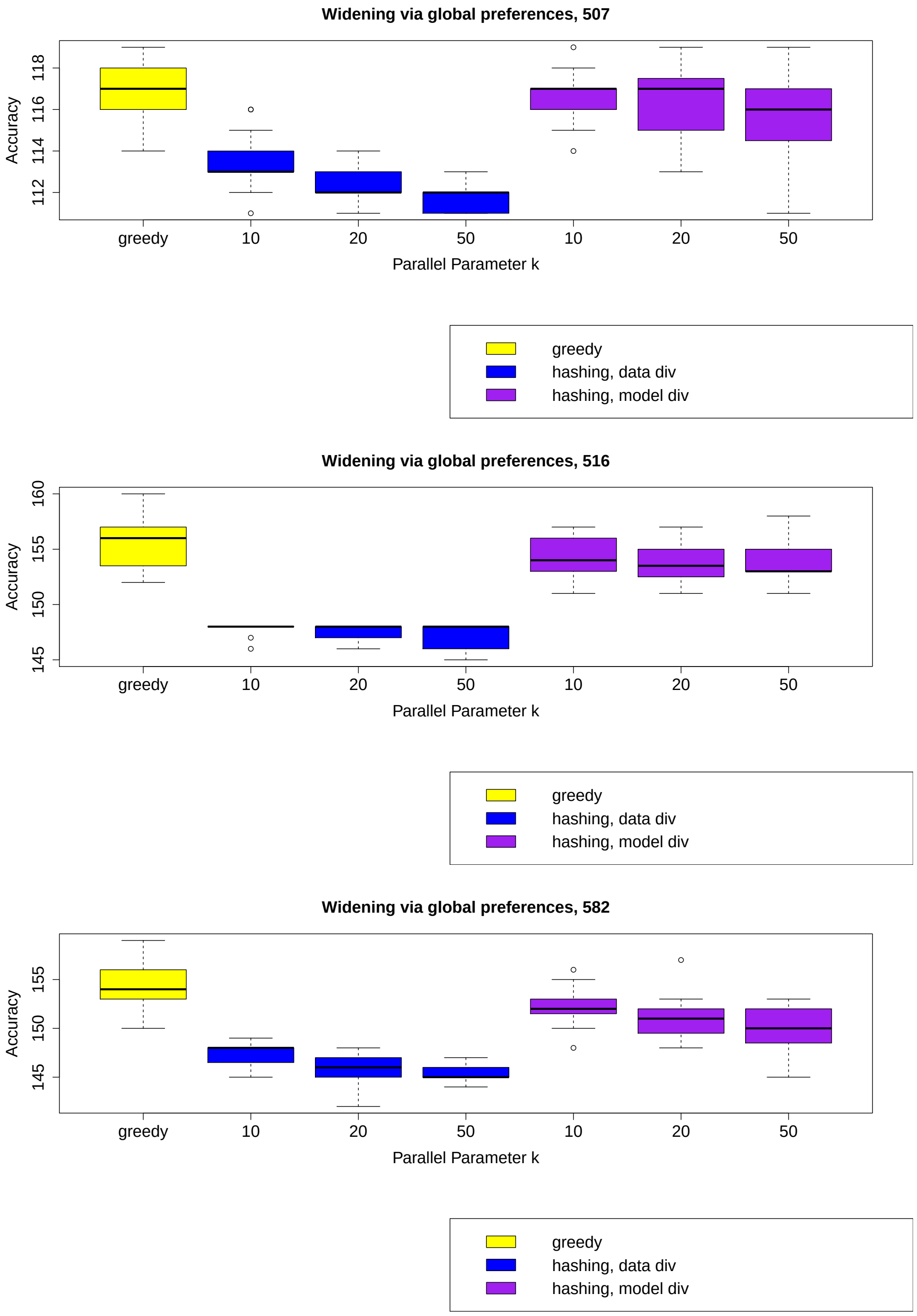


Figure 7.1: Results from the evaluation of $Top-k$ Widening with and without diversity.

7.6.2 Communication-less Widening via Hashing (Global Preferences).

By varying parameter t, we can control how much the selection paths of the parallel workers deviate from the selection path explored by the greedy SCP algorithm. The


Widening via global preferences, 507

Widening via global preferences, 516

Widening via global preferences, 582

Figure 7.2: Results from Widening via global preferences, with data-driven and model-driven diversity for $t = 1$.

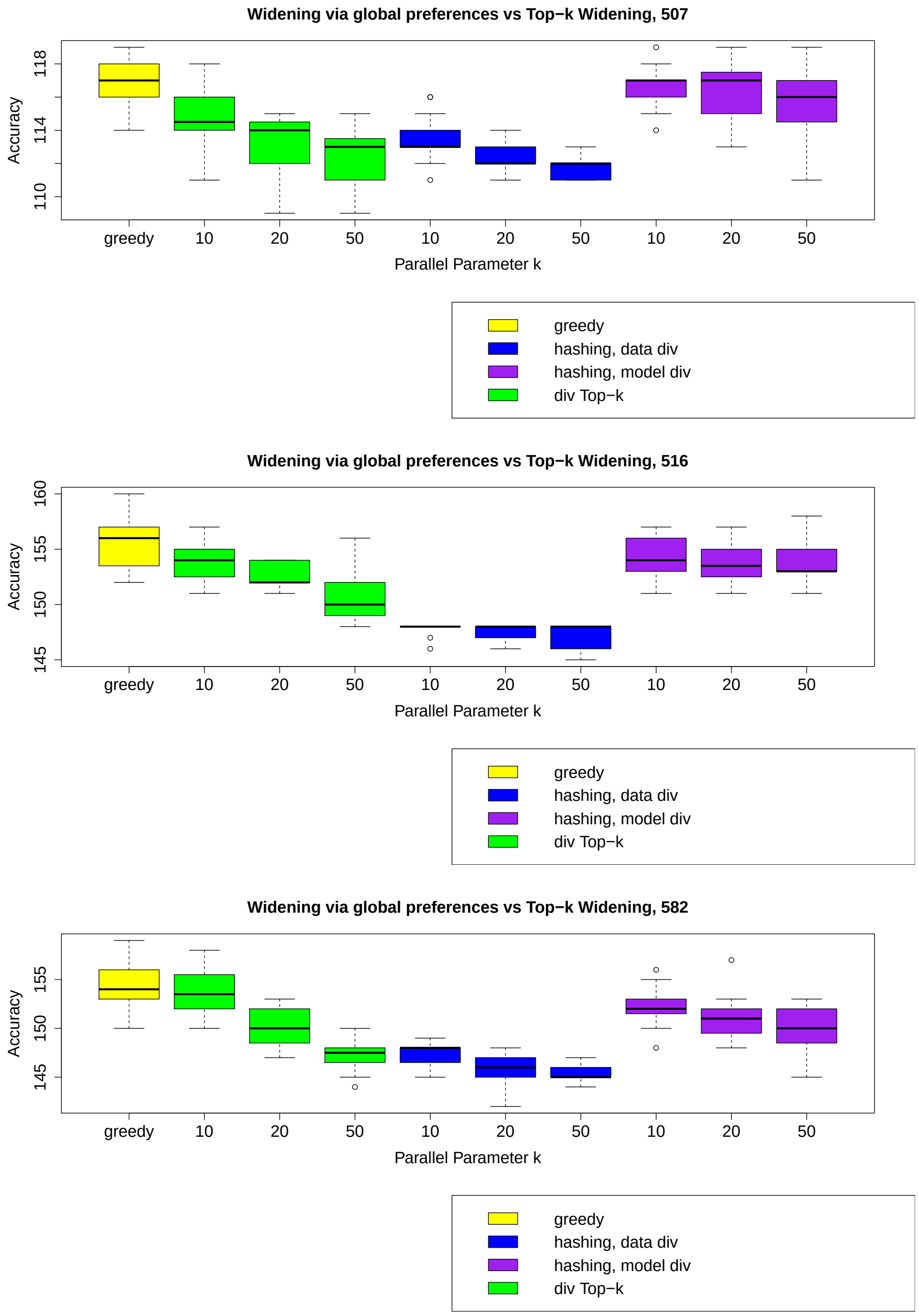


Figure 7.3: Results from Widening via global preferences for different diversity approaches contrasted with diverse $Top - k$.

parameter t controls the relative importance of the factors quality and diversity. For parameter $t \leq 1$, the parallel workers explore different selection paths of the greedy algorithm, only considering different paths that have equally good, local quality. Here the different orders π_i only serve for tie breaking. For parameter values $t > 1$, the selection paths of the parallel workers also include locally sub-optimal solutions. Large values of t ($\gg 1$) will lead to randomized exploration of the search space. Figure 7.2 shows the results for communication-free Widening with data- resp. model-driven diversity enforcement, while Figure 7.3 compares the communication-less approach with diverse $Top - k$ Widening. From the above results two main trends become clear. As expected, a larger width of the search improves the quality of the solution. Enforcing diversity improves the results even further. For communication-free Widening, the first set of tests simply enhances the greedy algorithm by exploring different options when breaking ties in-between equally good intermediate solutions. By increasing parameter t the widened algorithm is allowed to also explore paths of non-locally optimal choices, which further improves the results. The optimal value for parameter t depends heavily on the data set, and if fine-tuning is applied, more improvement can be expected. Obviously, if t is too large, this will turn the algorithm into an almost data-independent, random search process, deteriorating solution quality again. The Widening via hashing approach is comparable and in some cases performs better than the diverse $Top - k$ Widening (for the two fixed parameters). Note, that if one fine-tunes the diverse $Top - k$ with an appropriately selected parameter, it will outperform the hashing approach.

7.6.3 Widening via Optimality θ, k-neighborhoods.

Due to the Widening via neighborhoods being a sparse method, we can see that increasing θ can lead to worsening of the performance in certain. A very large neighborhood size θ leads to a randomization of the search. It is clear that the larger the number of parallel workers for a fixed neighborhood, the better the performance. For a fixed number of parallel workers, increasing the size of the neighborhood eventually will lead to a randomized search. A small size of the neighborhood leads to exploring solutions, which are similar. The optimal size of the neighborhood is dependent on the properties of the data. It is better to enforce diversity explicitly, instead of depending on the size of the neighborhood to introduce diversity.

7.6.4 Widening via Diverse Neighborhoods

The goal of Widening is not randomized diverse exploration, but exploration of the peaks (diverse and promising solutions) of the search space landscape. This is why diversity needs to be explicitly enforced, instead of relying on a very large neighborhood size. The results presented in Figure 7.6 show the performance of Widening via diverse neighborhoods for different values of the diversity threshold δ. The optimal value of the

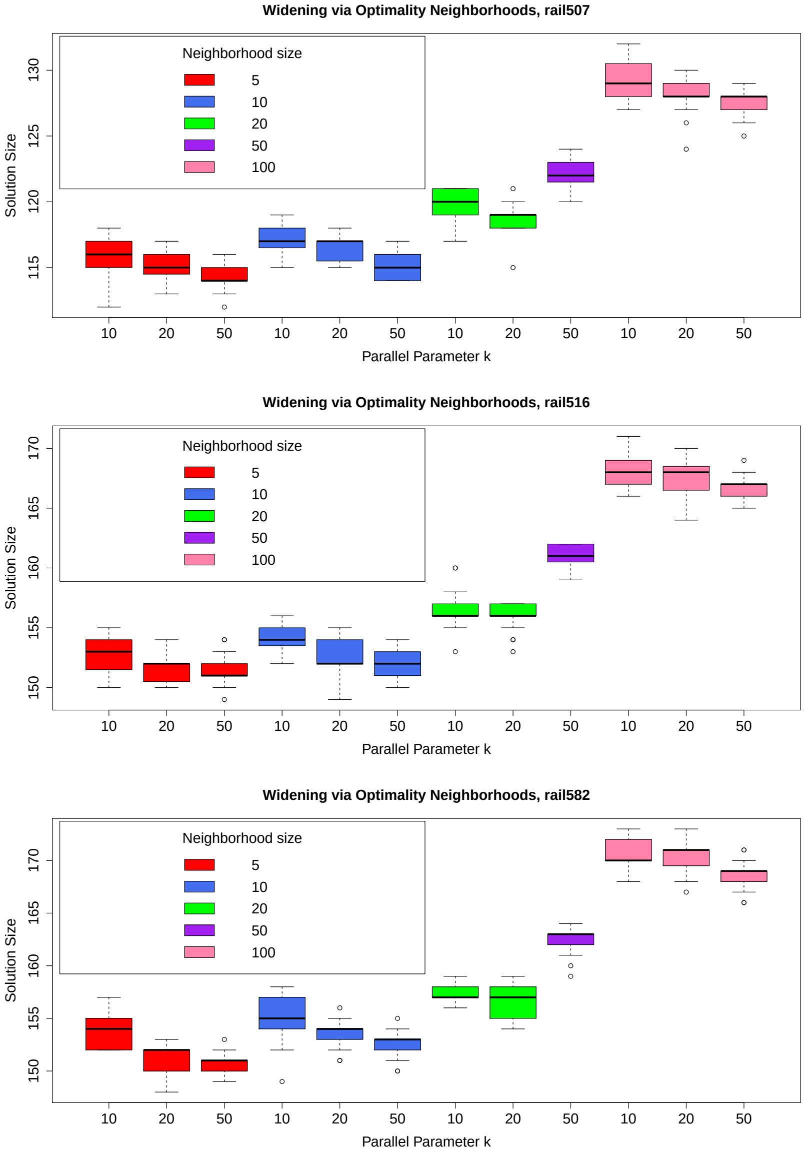


Figure 7.4: Widening via optimality neighborhoods, for different values of Widening parameters θ, k.

threshold will depend on the structure of the data/search space. The goal is to discover different, but optimal peaks, and thus the "peak distribution" will determine the optimal threshold. Furthermore, this optimal value may not be constant during the search, and it is hard to guess it apriori. This inflexibility is the main drawback of this method. The results for Widening via diversity neighborhoods, where diversity is obtained via fitness sharing, for varying values of the fitness sharing parameters are presented in Figure 7.7. Clearly, there are improvements by using Widening via diverse neighborhoods built by fitness sharing in two data sets *rail*507 and especially *rail*582, however, no benefits are noticed for *rail*516. It is difficult to tune well the fitness sharing parameters α, σ for a given data set.

7.6.5 Widening via Similarity Neighborhoods

Exploitation via Widening via Similarity Neighborhoods

Widening via similarity neighborhoods can be used for exploitation – an intensified detailed search within a region of the search space, which is already known for having good solutions. Pairwise Jaccard distance is used to assess similarity between two models in each experiment. Since we are widening the greedy algorithm, which is already known to produce a good solution, Widening via similarity neighborhoods aim at investigating the vicinity of the greedy algorithm in detail. The results of this exploitation are shown in Figure 7.8. As it is logical, the best performance for this Widening approach is achieved for a small neighborhood size and a large number of parallel workers. This leads to a more thorough investigation of a fixed region of the search space, and is in accordance with the theoretical results obtained in Chapter 5. For a fixed number of parallel workers, the smaller size of the neighborhood leads to improved performance. Inversely, a larger number of parallel workers, for a fixed neighborhood size improves the performance. Widening via similarity neighborhoods does not consider model performance, when selecting models at each step, only the similarity between a given model and a locally optimal model, that is why for a large neighborhood this leads to randomization of the search, instead of improvement.

Similarity Search using Widening via Similarity Neighborhoods

The second application of this type of Widening is a similarity search. We are looking for a set of models which are of high similarity among each other and of good performance.

Remark For this type of goal, we are interested in the properties of the full set of k final solutions and not only in the properties of the best solution from the final set of k solutions. This is why, we investigate the average similarity for the full set of final solutions, as well as the average performance of the set of final solutions.

In Figures 7.9, 7.10, 7.11 we can see that the larger is the size of the neighborhood θ, the smaller is the average pairwise similarity of the set of models. For a fixed neighborhood size, the greater number of parallel workers leads to a greater average pairwise similarity. We can see that for a very small neighborhood size $\theta = 2$ and high number of parallel workers $k = 100$ the method has the highest average similarity and the highest average performance.

In addition, we also compare the set of final k solutions obtained by Widening via similarity neighborhoods to the set of k solutions obtained by Widening via optimality neighborhoods, without the use of diversity. In Figure 7.12 the similarity of the models obtained by Widening via similarity neighborhoods is compared to the similarity of the set of models obtained by Widening via optimality neighborhoods. The set of models discovered by Widening via similarity neighborhoods has higher average pairwise similarity than the models discovered by the other type of neighborhoods, optimality neighborhoods. The difference is more pronounced the larger the neighborhood θ in comparison to the number of parallel workers. This is explained by the fact that for a small neighborhood based on either similarity or optimality, a small part of the search space is explored and as a consequence, the discovered models are similar. For large number of parallel workers, a larger portion of the search space is explored in the case of optimality neighborhoods, and a smaller part of the search space is explored but in greater detail (exploitation) by the parallel workers in Widening via similarity neighborhoods. The results also show that Widening via optimality neighborhoods will benefit from the use of diversity, especially for a small size of the neighborhood.

The best use of Widening via similarity neighborhoods, is to use more parallel workers in a smaller sized neighborhood. For further improvements, "good" solutions can be used as starting points, and from there look for better/optimal solutions in the vicinity of those good solutions, also referred to as exploitation. A similarity search can be performed by selecting apriori known desired properties, and models similar to those prerequisites, instead of choosing models similar to the greedy choice.

7.7 Runtime Analisys of Widening Approaches with and without Communication.

In this section, we will discuss the runtime of different Widening approaches and compare and contrast the Widening approaches with and without communication.

7.7.1 Hashing vs $Top - k$ Widening Approaches.

The experiments above were primarily concerned with evaluating and comparing properties of the solutions, such as model quality and similarity. However, we are also interested in preserving the running time of the widened algorithm equal to (or at least close to) that of the original greedy algorithm. Requiring frequent communication to find the top k solutions will become a bottleneck as k increases. What is of interest is, how big the differences in the running time are as more resources are available, which is why we are so interested in communication-less diverse subset selection. We used the $Top - k$ Widening method and contrasted it to data-based hashing (). The experiment were performed using the $rail507$ data set on a 64-core machine and repeated 10 times. Figure 7.13 displays the runtime for the different methods against an increasing number of parallel workers.

7.7.2 Widening via Neighborhoods vs $Top - k$ Approaches.

We used the $Top - k$ Widening method and contrasted it to the different neighborhood-based approaches. The experiment were performed using the $rail507$ data set on a 64-core machine and repeated 10 times. Predictably, the running time of the Widening approaches differ depending on the type of neighborhoods used. Widening via pure optimality neighborhoods is close to constant. The running time of Widening via similarity neighborhoods is not affected strongly by the size of the neighborhood used, but depends more on the size of the data and the total number of refinements and does not depend strongly on the number of parallel workers. This is because in order to select the θ most similar neighbors all refinements must be compared to the optimal refinement at each step. This can be seen in Figure 7.15, where for different sizes of neighborhoods the running times are not drastically different. This depends on the data set – for example, the data set $rail507$ contains many subsets (or 63,009 model fragments) and at each step the locally optimal model is compared to all $63,009 - l$ model fragments, where l is the refinement level.

There are two factors with similar effect in Widening via diverse neighborhoods, which determine the running time. One is the size of the neighborhood, the other is the diversity threshold. From Figure 7.14 it is clear that the size of the neighborhood is the most influential factor for the runtime of Widening via diverse neighborhoods. The greater the size of the neighborhood the more computationally intensive is the building of the neighborhood. However, the number of parallel workers does not influence the running time much, given a sufficiently high number of parallel resources. The size of the threshold does influence the running time, because a greater threshold implies more comparisons (due to the fact that some potential members of the neighborhoods will fail to meet the threshold and thus a greater number of comparisons will be required).

Additionally, Widening via similarity neighborhoods, for this particular data set and setting, as well as sizes of neighborhoods, has a worse running time than Widening via diverse neighborhoods because of the very large number of model fragments, and the requirement that at each step the similarity between the most optimal model and all other possible refinements is evaluated, while depending on the value of the threshold parameters, the comparisons required to build a diverse neighborhood at each step, may be significantly less for a fixed neighborhood size θ. Clearly, however, the communication between parallel workers remains the greatest bottleneck for the running time, as Figure

7.16 shows. The experiments show that methods which use communication have worse running time in comparison to methods which do not require communication between the parallel workers.

However, in Section 7.6.4, Figure 7.5 we saw that even for smaller sized neighborhoods the results from Widening via diverse neighborhoods are of comparable quality to Widening using communication. The diversity can be controlled through a larger threshold instead of a larger neighborhood size in order to have a less negative impact on the running time.

Because building the neighborhoods is the most computationally intensive aspect of the search, preprocessing can benefit both similarity and diversity searches, based on neighborhoods built in the refinement sets.

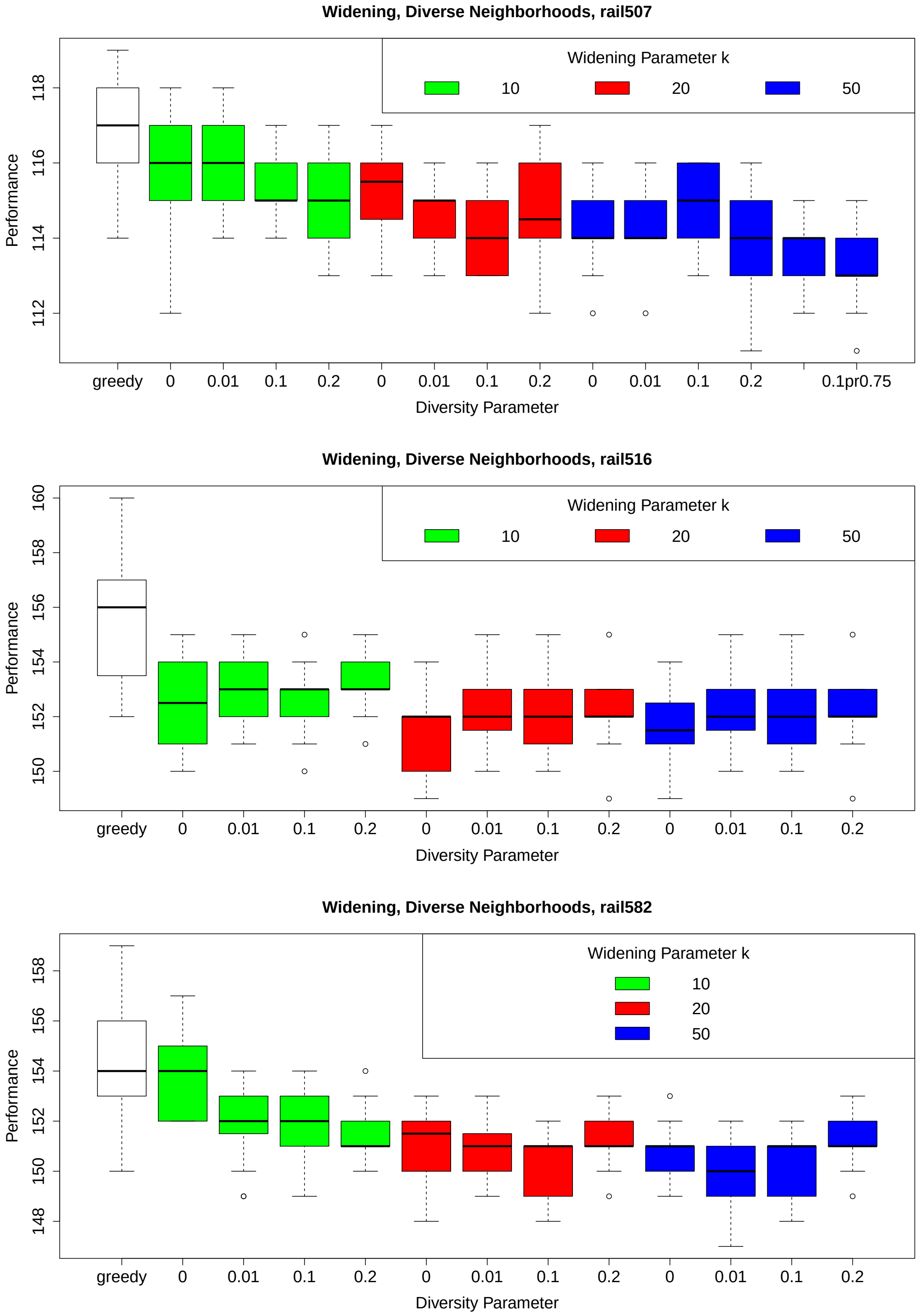


Figure 7.5: Widening via diverse neighborhoods for different values of Widening parameter k, for different values of the diversity threshold δ, and a fixed neighborhood size $\theta = 5$. Jaccard distance was used.

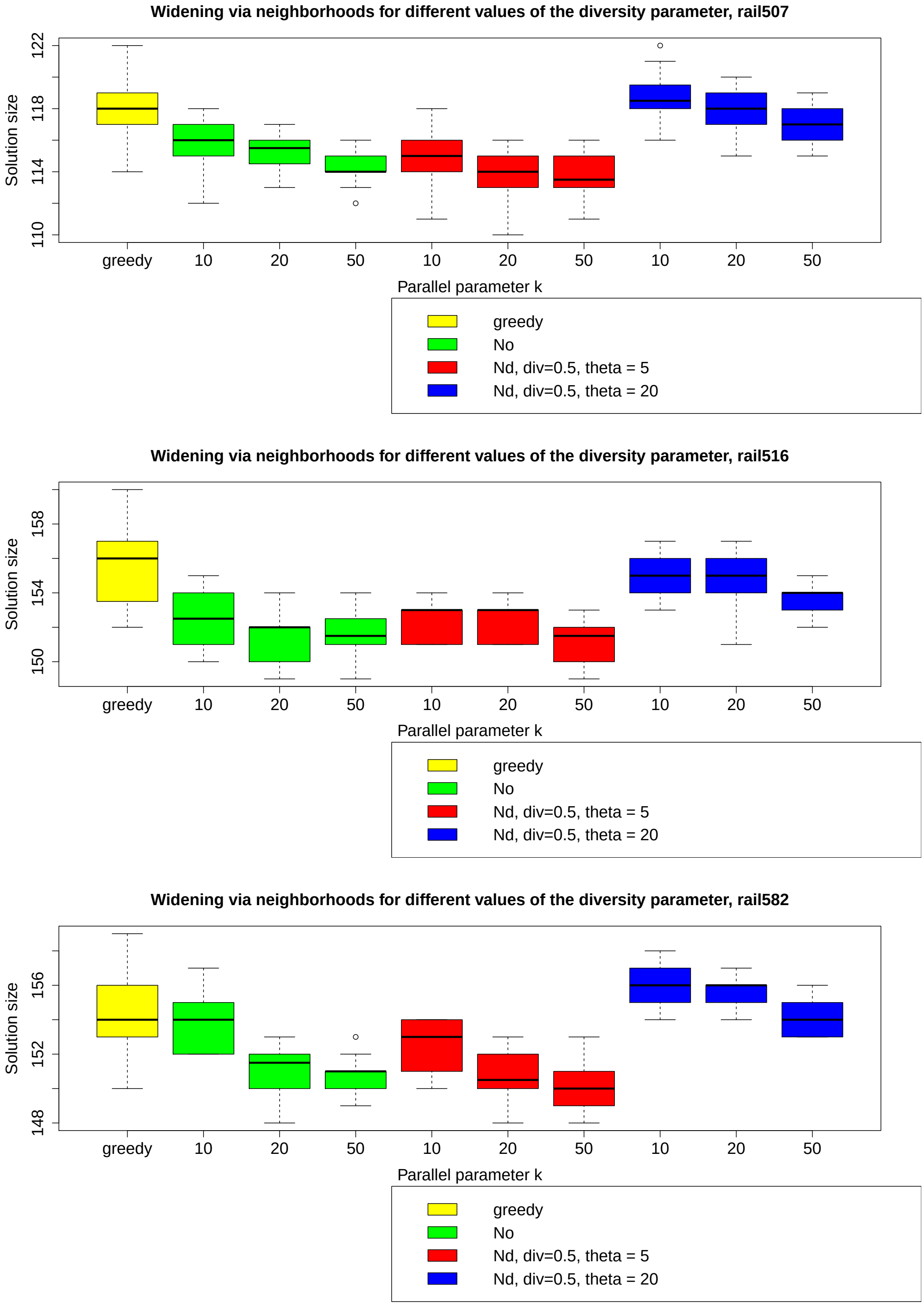


Figure 7.6: Widening via diverse neighborhoods for different fixed values of the diversity threshold δ, for different values of the Widening parameter k, and neighborhood size θ. Jaccard distance was used.

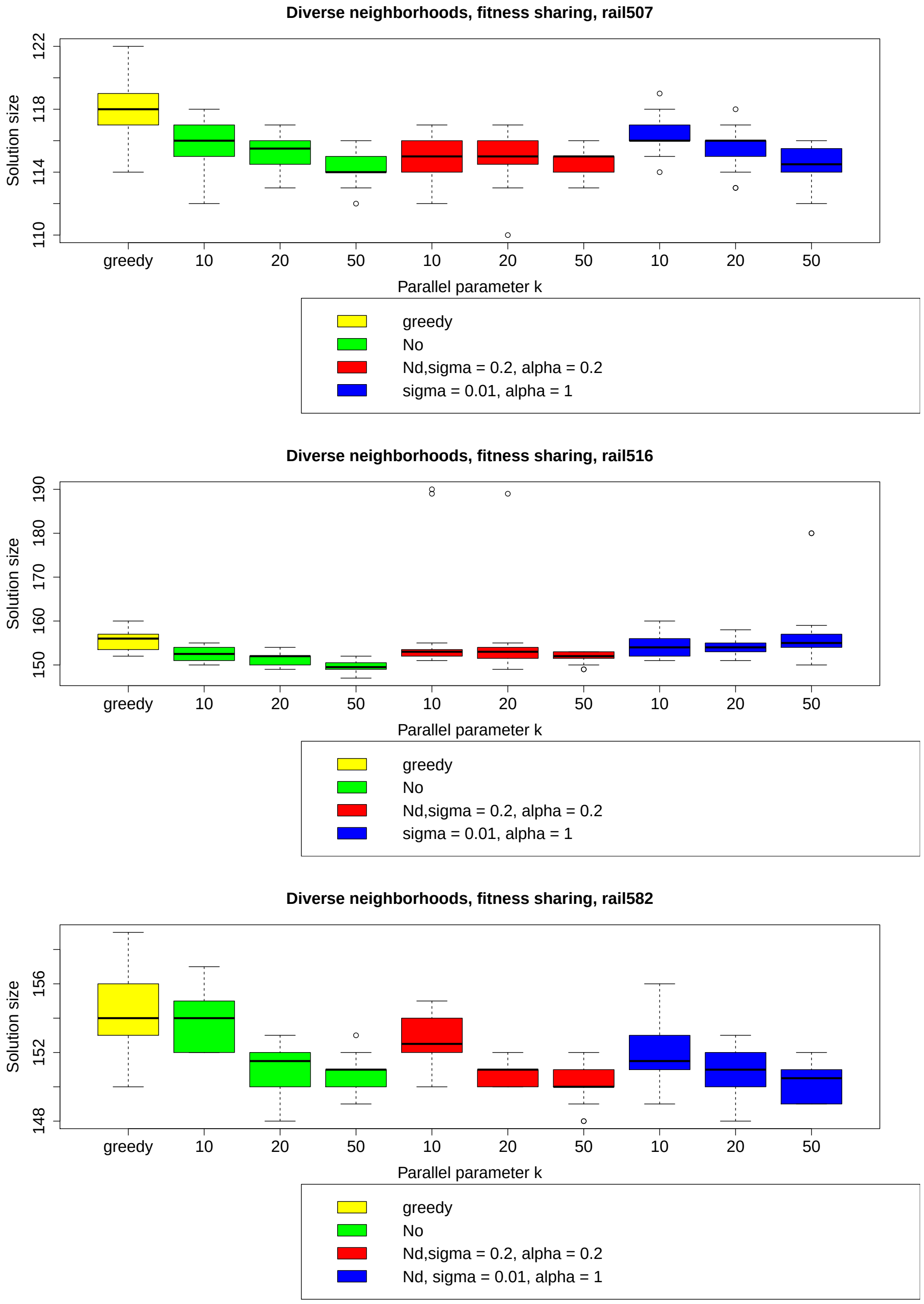


Figure 7.7: Widening via diverse neighborhoods, diversity achieved via fitness sharing. The effect of different values of the Widening parameters θ, k and different values of fitness sharing parameters α, σ for the different data sets are displayed.

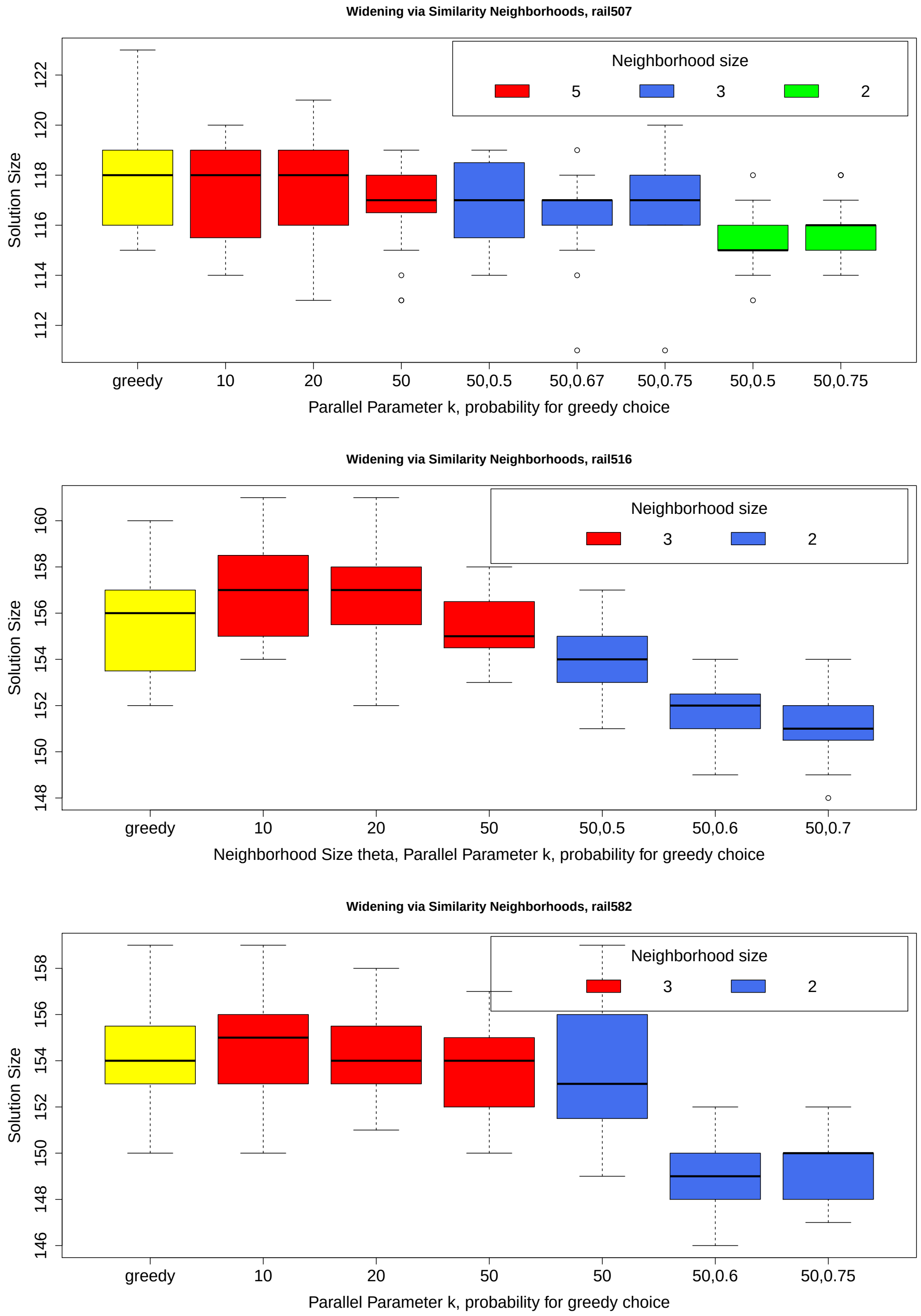


Figure 7.8: Widening via similarity neighborhoods for different values of θ. The similarity is evaluated by pairwise Jaccard distance.

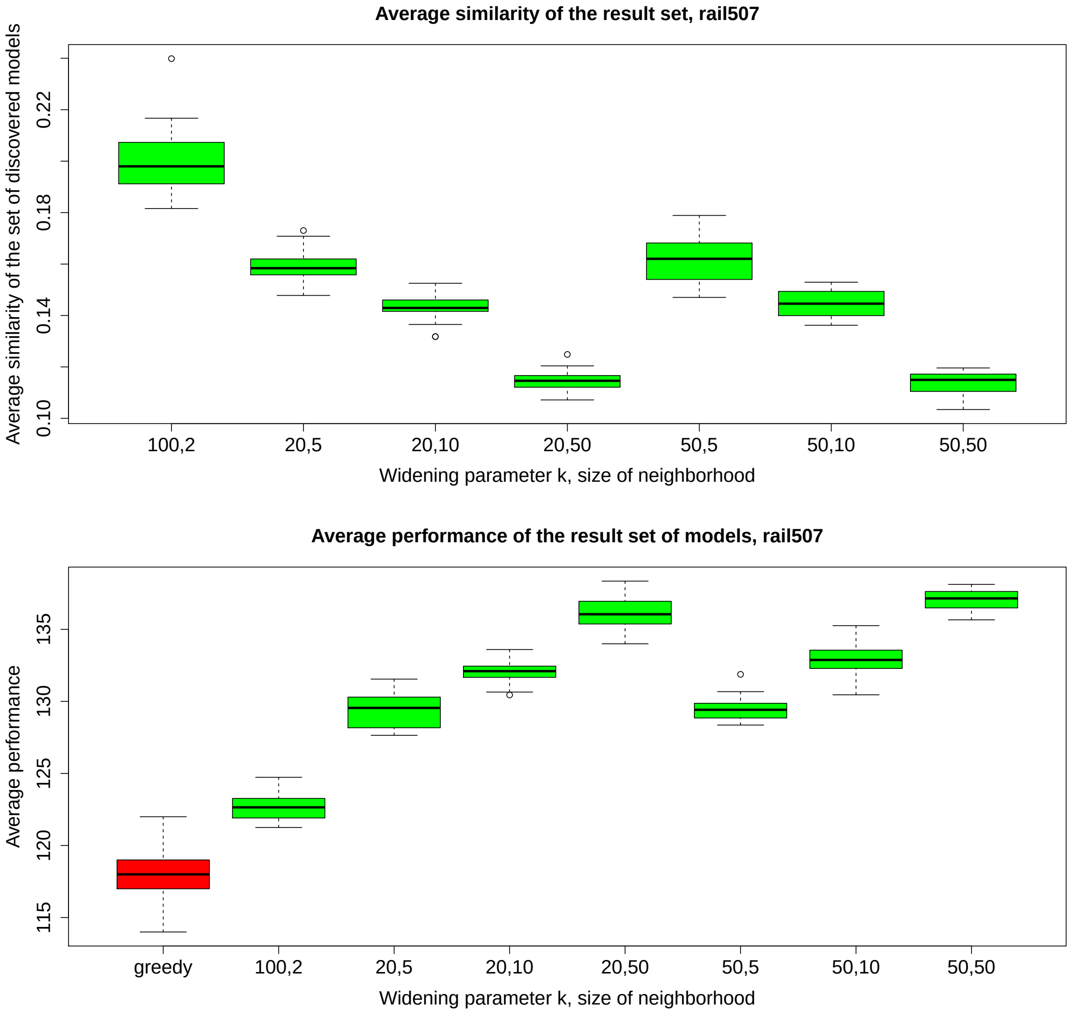


Figure 7.9: Average similarity and average performance of the discovered sets of models by Widening via similarity neighborhoods for the data set *rail*507. The similarity is evaluated by pairwise Jaccard distance.

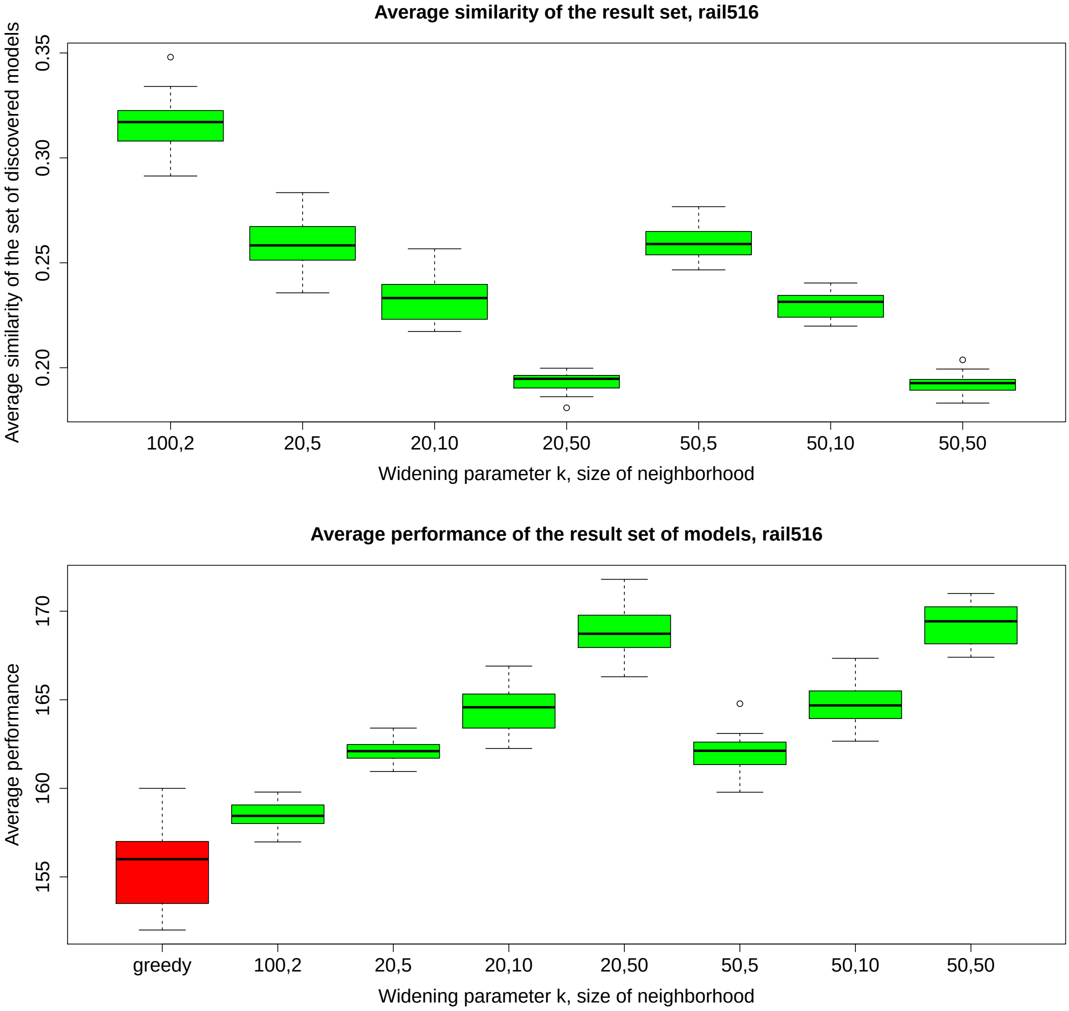


Figure 7.10: Average similarity and average performance of the discovered sets of models by Widening via similarity neighborhoods for the data set *rail*516. The similarity is evaluated by pairwise Jaccard distance.

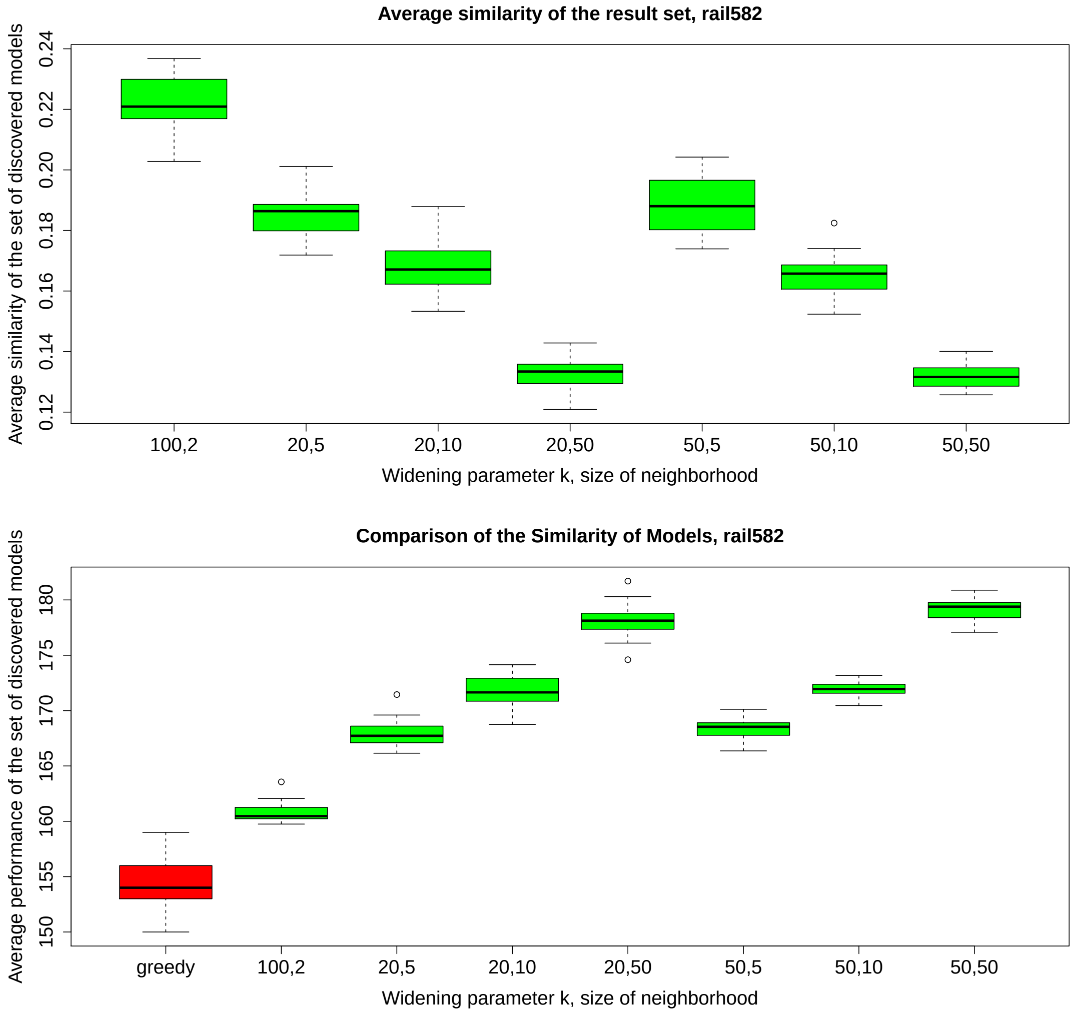
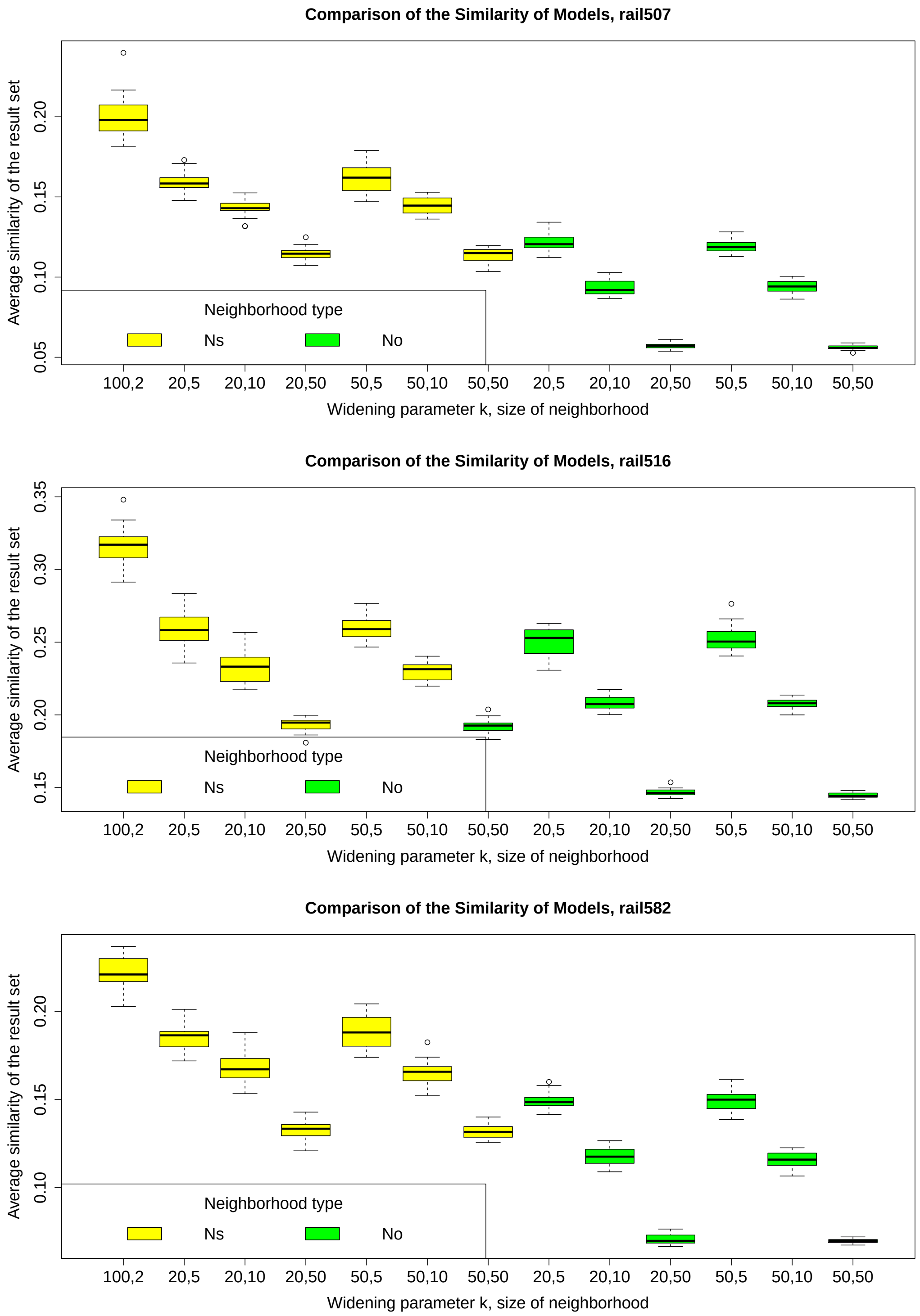


Figure 7.11: Average similarity and average performance of the discovered sets of models by Widening via similarity neighborhoods for the data set *rail*582. The similarity is evaluated by pairwise Jaccard distance.


Comparison of the Similarity of Models, rail507

Comparison of the Similarity of Models, rail516

Comparison of the Similarity of Models, rail582

Figure 7.12: Comparison of the average pairwise similarity of the set of discovered models for N_k^s and N_k^o.

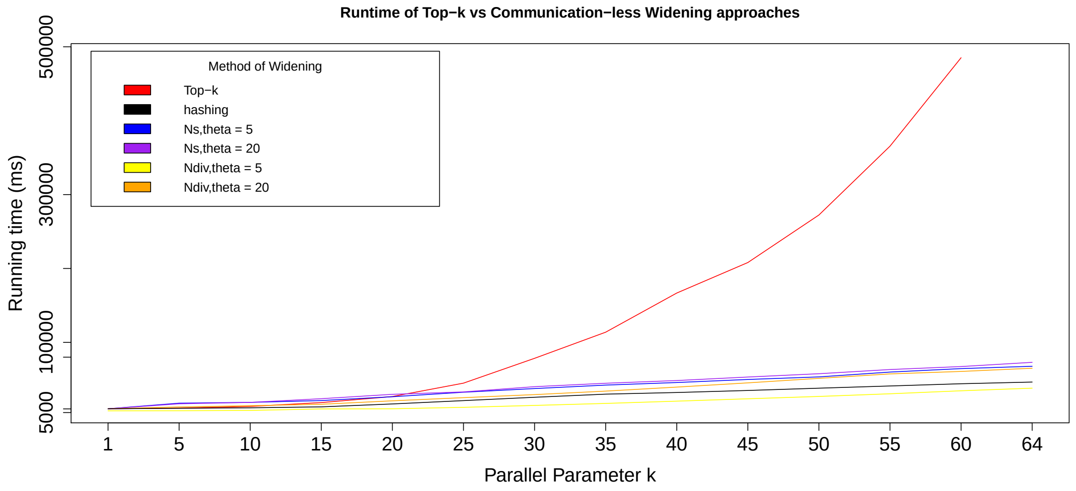


Figure 7.13: Results from the evaluation of the run-time for Widening of the greedy algorithm for SCP with and without communication (via global assignment of preferences and neighborhoods).

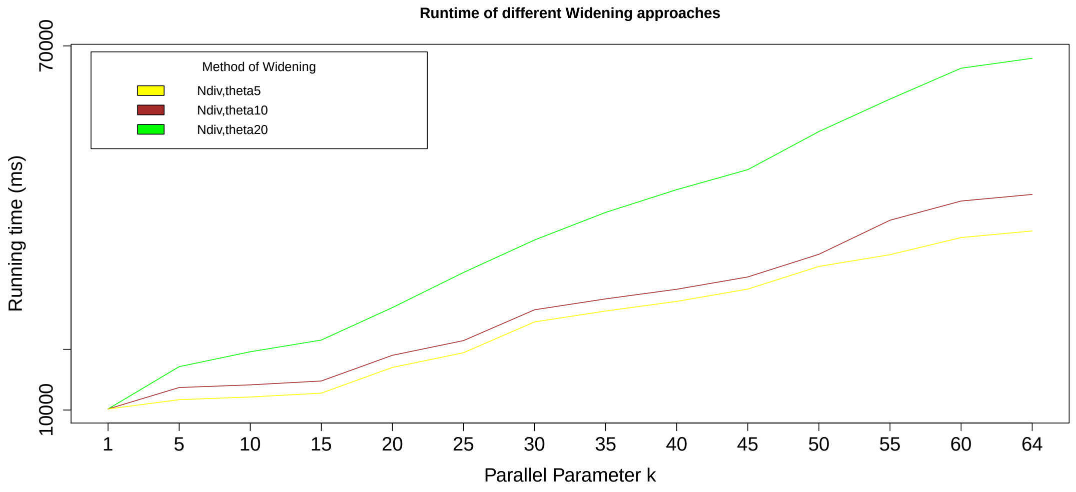


Figure 7.14: Comparison of the runtime of Widening via diversity neighborhoods for different values of parameter θ, neighborhood size, using $rail507$ data set. The size of the neighborhood is the most influential factor on the running time.

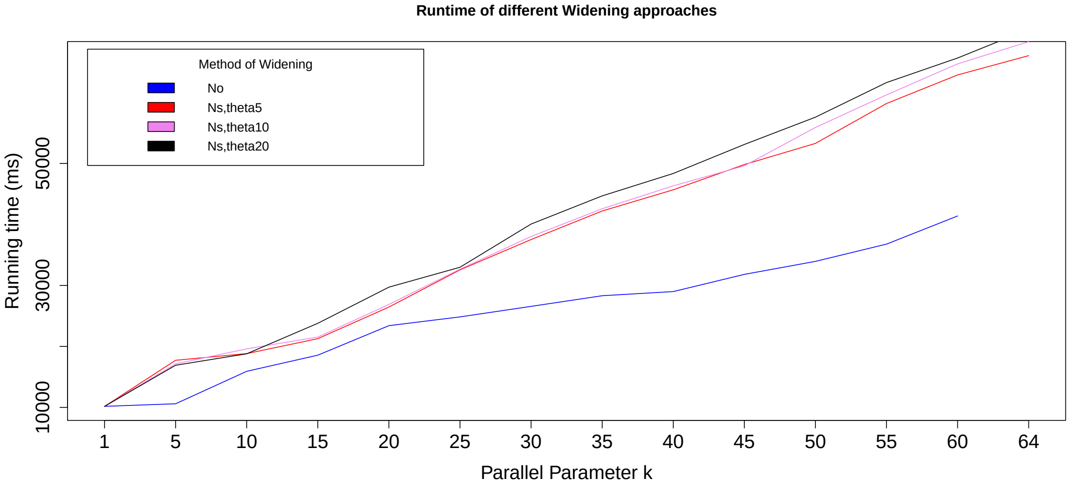


Figure 7.15: Comparison of the running time of Widening via similarity neighborhoods, for different values of parameter θ, neighborhood size, using $rail507$ data set. The size of the neighborhood does not have such a strong influence on the running time as with the Widening via diversity neighborhoods.

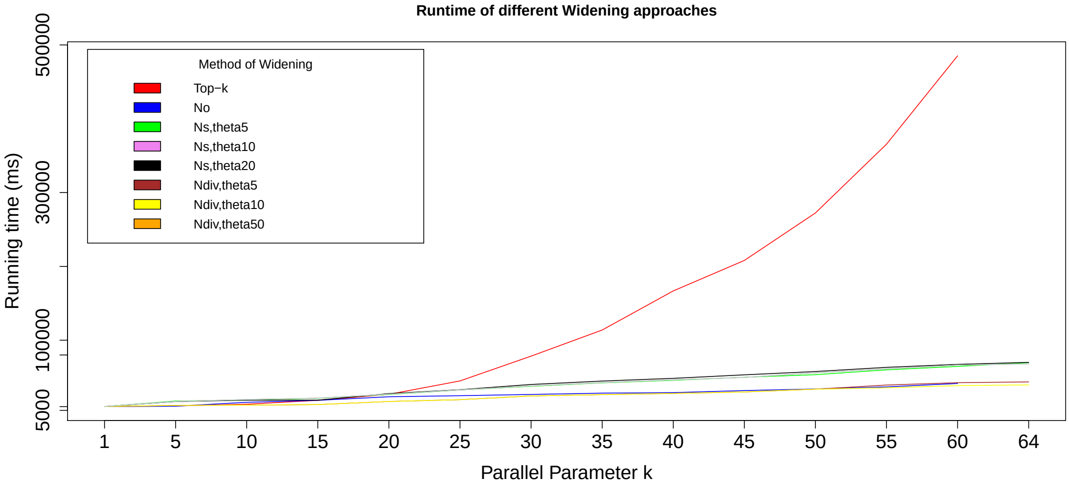


Figure 7.16: Comparison for the runtime of different Widening methods, with and without communication using $rail507$ data set.

Chapter 8

Widening of Rule Induction

This chapter is adapted from already published results from [87].

8.1 Decision Rules

Rule induction is an important machine learning technique, based on deriving formal rules from data. Rule induction, based on deriving rules from examples, is one of the fundamental methods in data mining. Decision rules are intuitive, compact, and interpretable. They can be easily used to express important characteristics from the data, and one can encode background knowledge in them[77].

Decision rules can be expressed as follows:

$$if(attribute_1, value_1)and \cdots and(attribute_n, value_n)then(decision, value)[77].$$

More complex rules are also possible. A large variety of approaches to decision rule induction exist. These include LEM1, LEM2, AQ, TDIDT and others, for a detailed description, refer to [77]. In this chapter, we will focus on the CN2 algorithm, which integrates strategies from both AQ and TDIDT. The original AQ algorithm has difficulties with noise handling, which is why the CN2 enhances it by either uses extra evaluation criteria or pruning. A description of the algorithm and its benefits follows below.

8.1.1 CN2 Algorithm

The original CN2 algorithm results in an *ordered list* of rules. However we will use a version of $CN2$, which results in *unordered set of rules*, due to the fact that it is easier to widen and also offers better interpretability. We use the version of the CN2 algorithm as described in [27], which results in an *unordered set of rules.*

```
Algorithm 3: Unordered CN2 Algorithm
Data: A set $E$ of classified examples, $classes$
Result: A list of rules $finalRuleset$ to classify data in one of $classes$
let $rulelist \leftarrow \emptyset$ ;
foreach $class \in classes$ do
    generate $rules$ by $CN2ForOneClass(examples, class)$ ;
    add $rules$ to $rulelist$ ;
return $rulelist$;
```

The CN2 algorithm utilizes a beam search, with size $MAXSIZE$, which can be viewed as several parallel searches. During the beam search, at each step the best $MAXSIZE$ candidates are selected for further investigation.

```
Procedure FindBestCondition(examples,class)
bestcondition $\leftarrow \emptyset$;
let $STAR = \{bestcondition\}$;
while $STAR$ is not empty do
    Specialize all complexes in $STAR$:
     $NEWSTAR \leftarrow \{xANDy : x \in STAR, y \in SELECTORS\}$;
    Remove all conditions in $NEWSTAR$ that are either in $STAR$ or null;
    foreach $C \in NEWSTAR$ do
        If $\psi(C) > \psi(bestcondition)$ then bestcondition $\leftarrow C$
    repeat
        Remove the worst condition from $NEWSTAR$
    until size of $NEWSTAR < MAXSIZE$;
    $STAR \leftarrow NEWSTAR$
return bestcondition;
```

The $s(r(m))$ iteration step here is presented as generating specializations of the temporary best condition (conditions in the case of beam search), and then selecting the best one(s), based on a quality measure. The algorithm is described in the pseudocode in Algorithm 3. Procedure FindBestCondition presents the part of the algorithm, which is relevant to Widening of the CN2 algorithm. For a detailed description of the algorithm, refer to [27].

8.1.2 Widening of the CN2 Algorithm

The original design of the CN2 algorithm already has the option of exploring several solutions in parallel by allowing for the use of *beam search*, when looking for the best condition. We will investigate how much changing the size of the beam ($STAR$) affects the accuracy and will compare our methods to this very simple "Widening", which is native to the unordered CN2 algorithm. Note that the original Procedure FindBestCondition returns only one best condition each time. Below we discuss how the different Widening approaches can be applied to the CN2 algorithm. The implementation of $Top - k$

```
Procedure FindTop-KBestConditions(examples,class)
  $k - bestconditions \leftarrow \emptyset$;
  let $STAR = \{k - bestconditions\}$ ;
  while $STAR$ is not empty do
      Specialize all complexes in $STAR$:
      $NEWSTAR \leftarrow \{xANDy : x \in STAR, y \in ATTRIBUTETESTS\}$;
      Remove all conditions in $NEWSTAR$ that are either in $STAR$ or null ;
      forall $C \in NEWSTAR$ do
          $min \leftarrow min_\psi(k - bestconditions)$;
          If $\psi(C) > \psi(min)$ then $k - bestconditions.delete(min)$;
          $k - bestconditions.add(C)$;
          $sort(k - bestconditions)$;
      repeat
          Remove the worst condition from $NEWSTAR$;
      until size of $NEWSTAR < MAXSIZE$;
      $STAR \leftarrow NEWSTAR$;
  return $k - bestconditions$
```

Widening involves a modification of the selection operator $s_{\text{Top-k-CN2}}$ to build k rule sets in parallel. Procedure FindBestCondition still employs the beam search, used in the original CN2, but is modified to return the top k best conditions discovered by the beam search at each step and add them to the k solutions (rule sets) that are being built. This can be viewed in FindTop-KBestConditions. In the case of the diverse $Top-k$ Widening, Jaccard distances are evaluated based on the examples covered and a threshold is used to maintain diversity in the set of the top k temporary solution candidates.

8.2 Communication-less Widening of CN2

8.2.1 Global Diversity Approaches Using Preferences.

In this Widening approach, k CN2-style searches are performed in parallel, each with beam $MAXSIZE = 1$ and individualized selection operator $s_i^{\text{hash-CN2}}$ with individualized model quality function ψ_i returning only one best condition. This approach is shown

```
Procedure WideningKPreferences(examples,class)
  $preferences \leftarrow generatepreferences()$ ;
  Start in parallel $k$ times;
  FindBestPref(preferences,examples,class,i);
  $i = 1 \ldots k$
```

```
Procedure FindBestPref(preferences,examples,class,i)
  $bestcondition \leftarrow \emptyset$;
  let $STAR = \{bestcondition\}$ ;
  while $STAR$ is not empty do
      Specialize all complexes in $STAR$:
       $NEWSTAR \leftarrow \{xANDy : x \in STAR, y \in ATTRIBUTETESTS\}$;
      Remove all conditions in $NEWSTAR$ that are either in $STAR$ or null ;
      forall $C \in NEWSTAR$ do
          $\psi'(C) \leftarrow \psi(C) + tpreferences[i](C)$ ;
          If $\psi'(C) > \psi'(bestcondition)$ then $bestcondition \leftarrow C$;
      repeat
          Remove the worst condition from $NEWSTAR$;
      until size of $NEWSTAR < MAXSIZE$;
      $STAR \leftarrow NEWSTAR$;
  return bestcondition;
```

in Algorithm WideningKPreferences. In the approach, which uses data-based diversity, different orders of preferences are assigned to the examples and the score of a given condition is decided based on the preferences of the examples it covers. In the model-based diversity approach, different orders of preferences are assigned to different attribute tests. The preference for a condition depends on the attribute tests it consists of. These individualized preferences should be different for each parallel worker, in order to assure that each explores a different search path. In order to balance between model quality and model diversity, the model evaluation function of the individualized selection operator $s_i^{\text{hash-CN2}}$ is:

$$\psi_i(\text{condition}) = \psi(\text{condition}) + tp_i(\text{condition}),$$

where t is a parameter which controls how much importance should be given to the quality and to the diversity. For a very small value of t, the diversity can be used only for tie-breaking between equally good options; for very high values, the quality of the model loses importance in the selection process, and the exploration of the search space becomes randomized. To maximize diversity ideally a set of orders on preferences that have maximal inversion distances, should be generated. However, this is computationally highly intensive, and that is why a "sufficiently diverse" set of preferences is generated.

8.2.2 Communication-less Widening via Neighborhoods

```
Procedure FindBestConditionN_k^o(examples,class,labels,i)
bestcondition ← ∅;
let STAR ← {bestcondition} ;
while STAR is not empty do
    Specialize all complexes in STAR:
      NEWSTAR ← {xANDy : x ∈ STAR, y ∈ ATTRIBUTETESTS};
    Remove all conditions in NEWSTAR that are either in STAR or null ;
    forall C ∈ NEWSTAR do
        If ψ(C) > ψ(bestcondition) then bestcondition ← C;
    repeat
        Remove the worst condition from NEWSTAR;
    until size of NEWSTAR < MAXSIZE;
    STAR ← NEWSTAR;
neighborhood ← generateOptimalityNeighborhood(STAR, θ);
neighbestcondN_k^o ← optimalityNeighbor(neighborhood, bestcondition, labels[i]);
return neighbestcondN_k^o;
```

In the Widening via neighborhoods k parallel searches are started independently, without communication. Each parallel worker performs the refinement operation, which

```
Procedure $FindBestConditionN_k^s$(examples,class,labels,i)
  $bestcondition \leftarrow \emptyset$;
  let $STAR = \{bestcondition\}$ ;
  while $STAR$ is not empty do
      Specialize all complexes in $STAR$:
        $NEWSTAR \leftarrow \{xANDy : x \in STAR, y \in ATTRIBUTETESTS\}$;
      Remove all conditions in $NEWSTAR$ that are either in $STAR$ or $null$ ;
      forall $C \in NEWSTAR$ do
          If $\psi(C) > n\psi(bestcondition)$ then $bestcondition \leftarrow C$;
      repeat
          Remove the worst condition from $NEWSTAR$
      until size of $NEWSTAR < MAXSIZE$;
      ;
      $STAR \leftarrow NEWSTAR$;
  $neighborhood \leftarrow generateSimilarityNeighborhood(STAR, bestcondition, \theta)$;
  $neighbestcondN_k^s \leftarrow neighbor(neighborhood, labels[i])$;
  return $neighbestcondN_k^s$;
```

```
Procedure $FindBestConditionN_k^d$(examples,class,labels,i)
  $bestcondition \leftarrow \emptyset$;
  let $STAR \leftarrow \{bestcondition\}$ ;
  while $STAR$ is not empty do
      Specialize all complexes in $STAR$:
        $NEWSTAR \leftarrow \{xANDy : x \in STAR, y \in ATTRIBUTETESTS\}$;
      Remove all conditions in $NEWSTAR$ that are either in $STAR$ or $null$ ;
      forall $C \in NEWSTAR$ do
          If $\psi(C) > \psi(bestcondition)$ then $bestcondition \leftarrow C$;
      repeat
          Remove the worst condition from $NEWSTAR$;
      until size of $NEWSTAR < MAXSIZE$;
      $STAR \leftarrow NEWSTAR$;
  $neighborhood \leftarrow$
   $generateDiverseNeighborhood(STAR, bestcondition, \theta, threshold)$
   $neighbestcondN_k^d \leftarrow neighbor(neighborhood, labels[i])$;
  return $neighbestcondN_k^d$;
```

```
Procedure generateDiverseNeighborhood(STAR,bestcondition,$\theta$,threshold)
  sort($STAR$);
  $i \leftarrow 1$;
  $neighborhood \leftarrow \emptyset$;
  $neighborhood.add(bestcondition)$;
  while $(neighborhood.size() < k)$ do
      if $JaccardDistance(STAR[i], bestcondition) \geq threshold$ ;
      then $neighborhood.add(STAR[i])$;
      $i++$;
  return neighborhood;
```

```
Procedure generateOptimalityNeighborhood(STAR,$\theta$)
  $sort(STAR)$;
  forall $(i = 1, \ldots, \theta)$ do
      $neighborhood.add(STAR[i])$;
  return neighborhood;
```

```
Procedure generateSimilarityNeighborhood(STAR,bestcondition,$\theta$)
  $neighborhood[k] \leftarrow \emptyset$;
  $PriorityQueue\ \ similarity =$
   $newPriorityQueue(STAR.size() - 1, DistanceComparator)$;
  $min \leftarrow jaccardDistance(bestcondition, STAR[1])$ ;
  forall $(i \in 1 : STAR.size())$ do
      $STAR[i].distance \leftarrow Jaccarddistance(STAR[i], bestcondition)$;
      similarity.add(STAR[i]);
  $neighborhood.add(similarity[1 : \theta - 1])$;
  return neighborhood;
```

builds all possible refinements. In the case of the CN2 algorithm, the refinement operation consists of adding a single attribute test to a model. Then each parallel worker identifies the locally optimal refinement (the greedy choice) from the set of all possible refinements. Each parallel worker then builds a neighborhood of size θ of the locally optimal model, where the neighbors are also refinements from the same refinement set. The selection operator of each parallel worker is modified to choose an apriori assigned neighbor as its best next attribute. The difference between the different approaches to Widening via neighborhoods is only in the type of neighborhood built, namely, optimality, similarity, or diversity. This difference is demonstrated in the different modifications of Procedure FindBestCondition, namely, Procedures $FindBestConditionN_k^s$, $FindBestConditionN_k^o$, $FindBestConditionN_k^d$. Similarity neighborhoods are built using the Jaccard distance, evaluated using the examples covered by each condition. The optimality neighborhoods are defined using the original evaluation performance ψ of the CN2, the *Laplacian error estimate*:

$$LaplaceAccuracy := \frac{n_c + 1}{n_{tot} + q},$$

where n_c is the number of examples in the predicted class c covered by the rule, n_{tot} is the total number of examples covered by the rule, and q is the number of classes in the domain.

8.3 Methods and Implementation.

The different approaches to Widening were implemented using KNIME [15]. Seven data sets with different properties were used in the evaluation, described in Table 8.1. Each experiment was repeated 50 times and different samples were chosen at random for training and testing each time. In each repetition, 90% of a given data set was sampled for training the models, and the remaining 10% was used for testing the accuracy of the obtained model.

8.3.1 Widening Approaches with Communication between Workers

The simplest way to "widen" the CN2 algorithm is to increase the beam size of the search, which the algorithm naturally uses. Both, $Top - k$ and simple beam are implemented. Diversity is added to the $Top - k$ Widening approach.

8.3.2 Widening via Optimality Neighborhoods.

The performance of the best solution discovered at each run is collected and all of them are plotted in a box plot. Different values for k and θ are used and their performance is compared in order to evaluate the effect that the different values of these parameters have on the performance of the approach. Widening via optimality neighborhoods is the same as that of the original algorithm, if we disregard the pre-processing, which consists of assigning to each model fragment a list of labels $\{v_0, \ldots, v_{k-1}\}$, where v_i specifies which neighbor of model m' will s_i choose. The values for v_i are chosen randomly without repetition from $\{0, \ldots, k-1\}$.

8.3.3 Widening via Similarity Neighborhoods

The goal of Widening via similarity neighborhoods is to exploit an area of the search space. The performance of the best solution discovered at each run is collected and all of them are plotted in a box plot. Different values for k and θ are used and their performance is compared in order to evaluate the effect that the different values of these parameters have on the performance of the approach. Additionally, we evaluate the similarity of the models that each run of Widening via similarity neighborhoods produces. Each run produces k solutions and the similarity between them is evaluated using the data-based Jaccard distance. Furthermore, the average performance of all the k models discovered in each run is also calculated. For this approach, we evaluate not only the best model of the N^s approach, but the whole set of obtained k models at each run.

8.3.4 Widening via Diverse Neighborhoods

This approach builds neighborhoods consisting of peaks, diverse and promising temporary solutions with high performance. The type of peak selection depends on the size of the neighborhood as well as the threshold used. Given a threshold δ and a neighborhood θ, θ highest peaks are chosen, at distance at least δ from each other. The data-based Jaccard distance measure is used to evaluate the distance between each model, when building the diverse neighborhoods. By limiting the size of the neighborhoods, θ, while still using diversity, the method focuses on strong peak selection.

The performance of the best solution obtained at each run is collected and all of them are plotted in a box plot. Different values for k, θ, and a threshold δ are used and their performance is compared in order to evaluate the effect that the different values of these parameters have on the performance of the approach.

data set	#attributes	#data items	#classes	distribution between classes
glass	10	214	6	imbalanced
pima	9	768	2	imbalanced
german	21	1000	2	imbalanced
bupa	7	345	2	balanced
haberman	4	306	6	imbalanced
ecoli	8	336	8	balanced
wine	13	178	3	balanced

Table 8.1: Table with properties of the data sets used for evaluation of the Widening approaches.

8.3.5 Runtime Evaluation

A 64 core machine was used for the runtime evaluations, from Amazon EC2 M4. Each method was run 10 times for each value of the parameter k and the average running time is plotted. With similarity and diverse neighborhoods different values of θ are used ($\theta = 5, \theta = 10, \theta = 20$). The threshold used for diversity is $\delta = 0.5$.

8.3.6 Data

To evaluate the different Widening approaches applied to the CN2 algorithm, we used 8 data sets, with different properties, described in table 8.1 from the UCI Machine Learning Repository [1].

8.4 Experimental Results and Discussion

In this section, we present and compare the experimental results of the different Widening methods applied to the CN2 algorithm.

Remark All the plots are made using R, [122]. The box plots visualize the performance of the Widening approaches. The black horizontal line is the median value. The bottom and top of the box are the first and third quartiles. For the ends of the whiskers the default positions, as defined in *boxplot.stats grDevices* from the R documentation, and are located at roughly a 5% and 95% of the confidence interval. Any data not included between the whiskers is plotted as an outlier with a small circle.

8.4.1 $Top - k$ Widening

In Figure 8.2 we compare $Top - k$ Widening to a simple increase in the beam size of the CN2 algorithm. The results show that simple increasing of the beam in CN2 search leads to a more modest improvement in comparison to $Top - k$. For larger k the results improve for both approaches. In contrast, actual $Top - k$ Widening leads to more significant improvement of the accuracy, due to the fact that k rule sets were built at the same time. Namely, the $Top - k$ Widening approach is better than simple beam search due to the additional exploration of partial solutions in parallel.

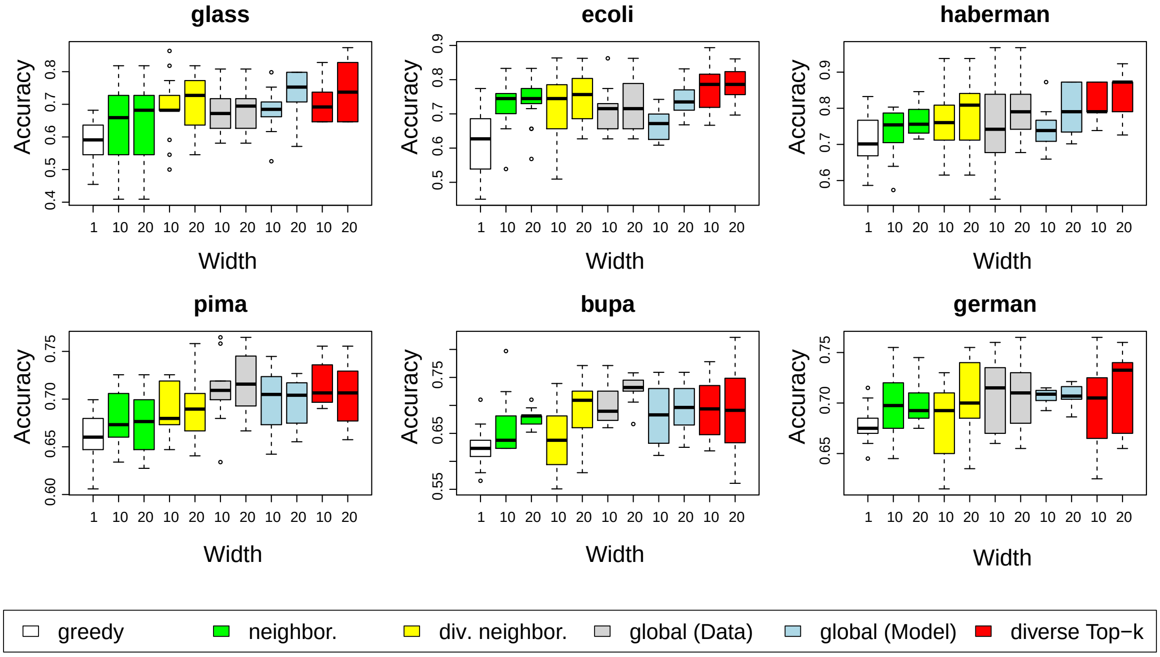


Figure 8.1: Experimental evaluation of different methods of communication-less Widening of the CN2 algorithm: Widening via optimality neighborhoods and Widening via assignment of global preferences compared with diverse $Top - k$ Widening on 6 different data sets.

Diverse $Top - k$ additionally improves the resulting accuracy, demonstrating that the use of diversity, when selected appropriately for the given data, improves the exploration of the search space. Communication-less approaches are contrasted with the one, which use communication in Figure 8.1. The results show that while using communication does produce better results, communication-less approaches are comparable with different $Top - k$ approaches.

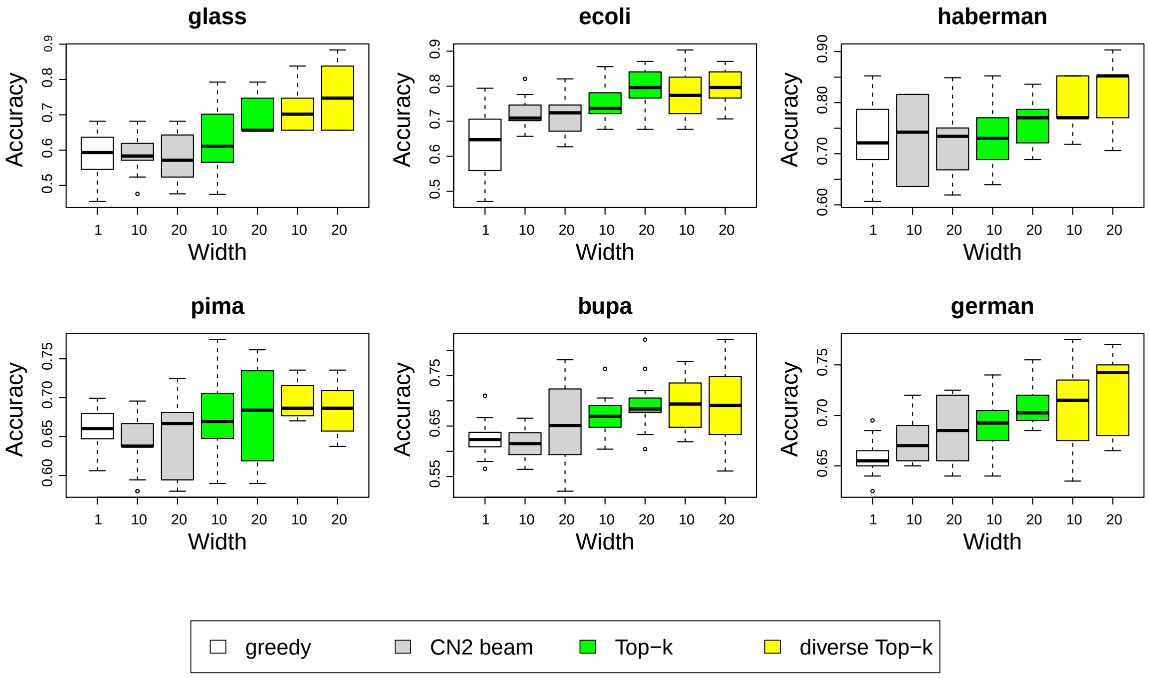


Figure 8.2: Evaluation of the unordered CN2 algorithm with different sizes of the beam, Top-k and diverse Top-k Widening of the CN2 algorithm on 6 different data sets.

Widening via global preferences achieves results comparable with approaches using communication, as shown in Figures 8.1, 8.3.

8.4.2 Widening via Optimality Neighborhoods

The performance of Widening via optimality neighborhoods is shown in Figures 8.4, 8.5. We can see that for larger k the results improve. Additionally, depending on the data set, different values for parameter θ are beneficial. The larger the neighborhood, the broader is the exploration of the search space. A smaller neighborhood size leads to an exploration of the search space closer to the greedy path. A larger number of parallel workers for a fixed neighborhood size leads to a better accuracy of the obtained model. The optimal size of the neighborhood is dependent on the data and the number of parallel workers available. A neighborhood size which is too large compared to the number of parallel workers, leads to randomized exploration. Inversely, a neighborhood size which is too small in comparison to the number of parallel workers will lead to exploring solutions which are too similar. In general, Widening via optimality neighborhoods needs to be paired up with diversity, in order to explore the search space in a good way. Simply

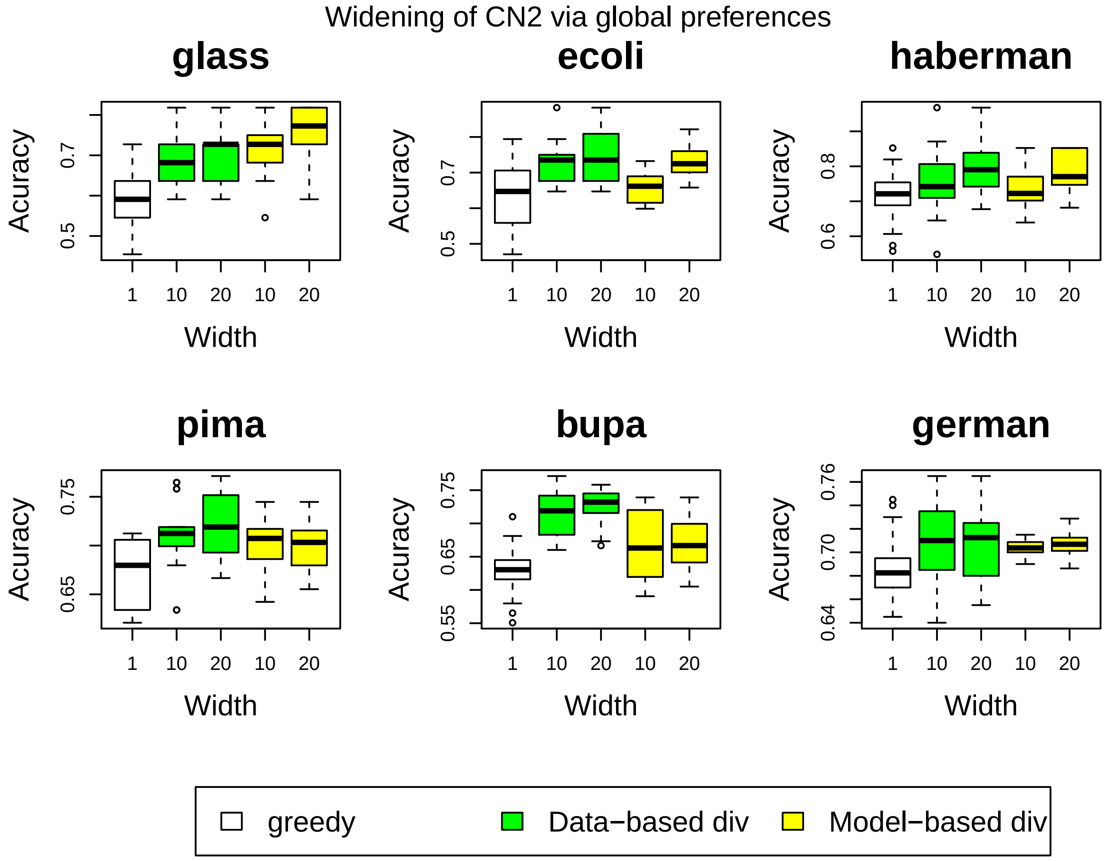


Figure 8.3: Widening of the CN2 algorithm for rule induction using global preferences.

increasing the neighborhood size, without explicitly enforcing diversity does not achieve the goal of Widening of exploring good and diverse solutions in parallel.

8.4.3 Widening via Similarity neighborhoods

8.4.4 Exploitation using Widening via Similarity Neighborhoods

One of the potential applications of the Widening via similarity neighborhoods is exploitation. For this we compare the quality of the solution obtained by this type of Widening to the one obtained by the greedy algorithm, as well as evaluate the similarity of the set of all θ models obtained at a given run. We can see in Figures 8.10, 8.11 that for a small neighborhood and a large number of parallel workers, the quality of the solution improves in comparison to the greedy one. Additionally, the higher the number of

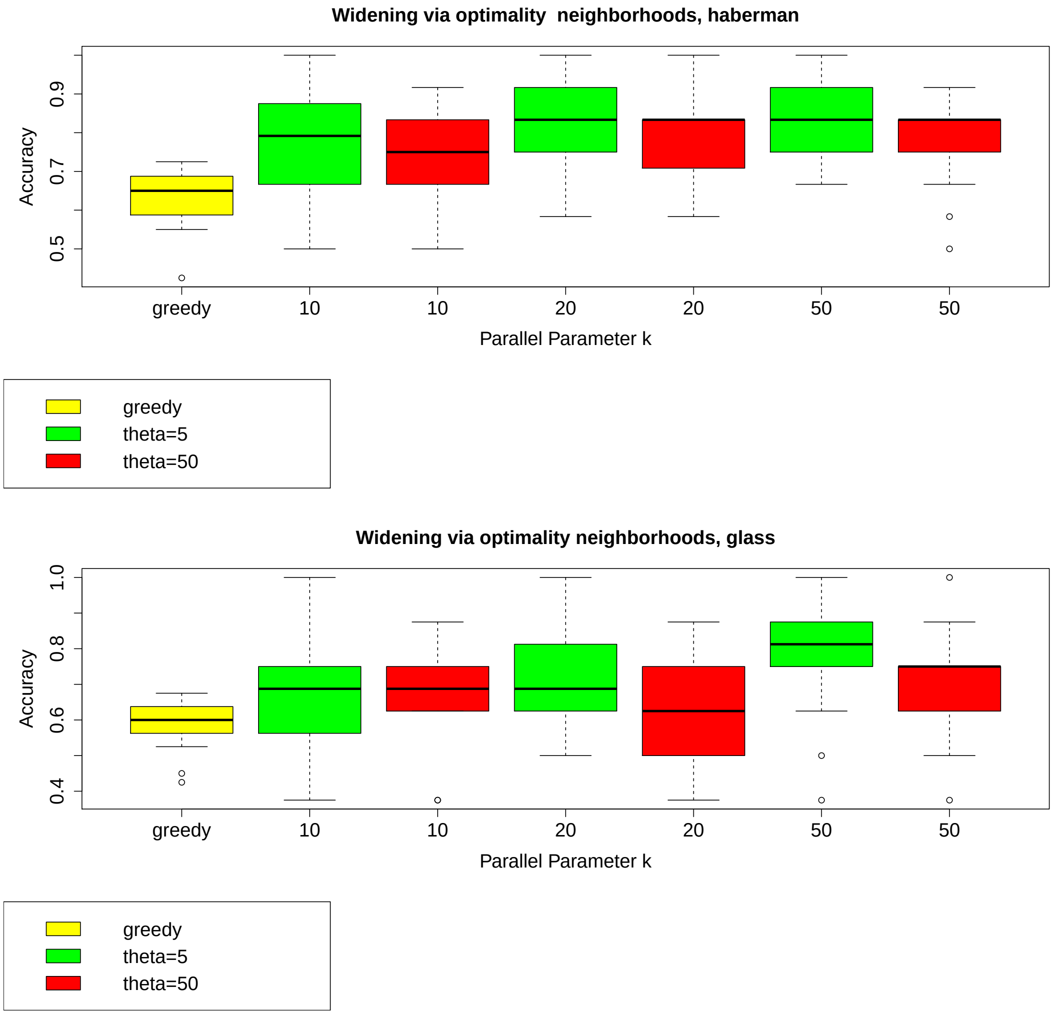


Figure 8.4: Performance of Widening via optimality neighborhoods on several data sets for different values of parallel parameter k and neighborhood size θ.

parallel workers and the smaller the size of the neighborhood, the higher is the average pairwise similarity of the set of models discovered. What can be concluded is that a higher number of parallel workers leads to a better performance of the method. Furthermore, different values of the parameter θ, the size of the neighborhood, is beneficial for different data sets. A small neighborhood size and a large number of parallel workers will lead to discovering a set of similar models of good performance.

A higher number of parallel workers for a fixed size of the neighborhood leads to a greater similarity. In order to improve the similarity of the set of models obtained by the method, the size of the neighborhood θ has to be small and the size of the number

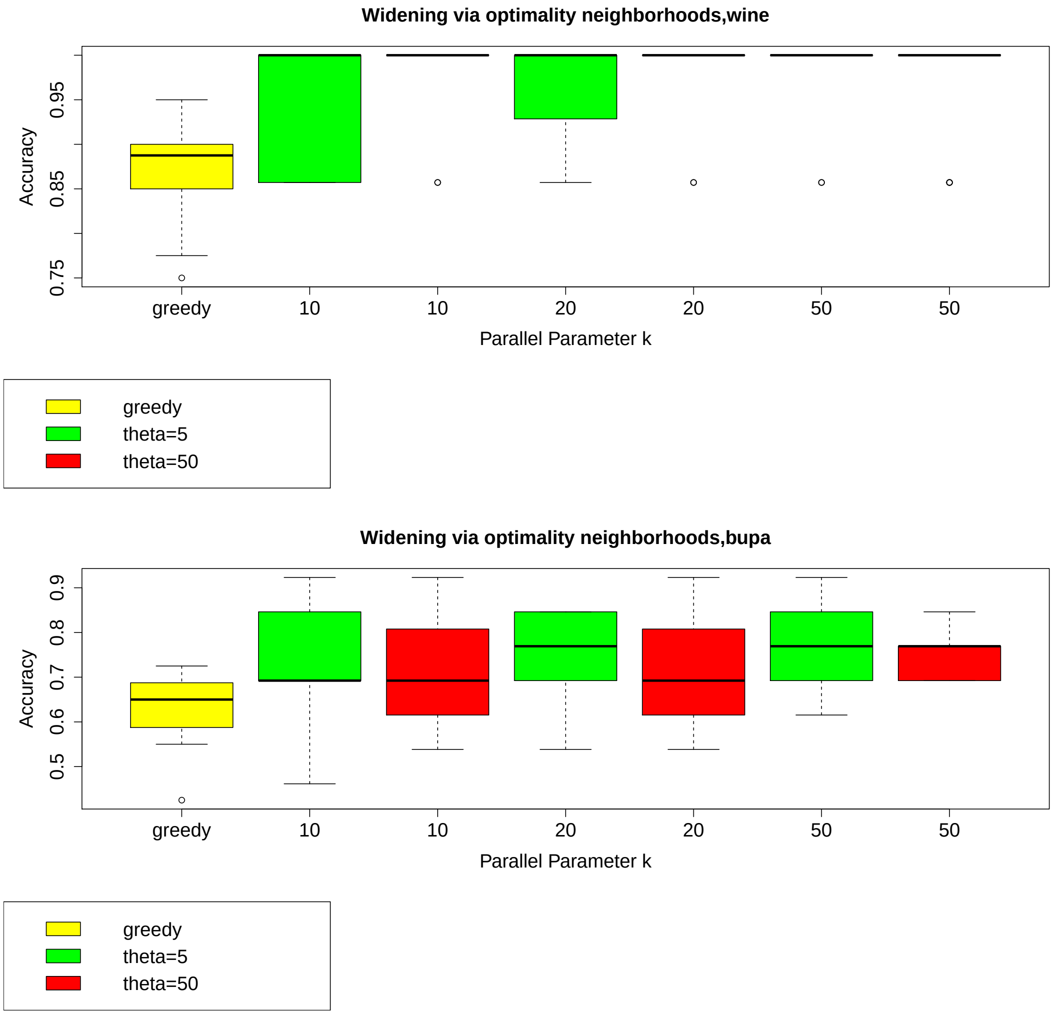


Figure 8.5: Performance of Widening via optimality neighborhoods on several data sets for different values of parallel parameter k and neighborhood size θ.

of parallel workers has to be large in comparison. A small neighborhood size leads to exploitation of a small region of the search space in the vicinity of the greedy solution. The average performance of the set of the models discovered by Widening approach $\{N_k^s\}$ is shown in Figures 8.9, 8.8.

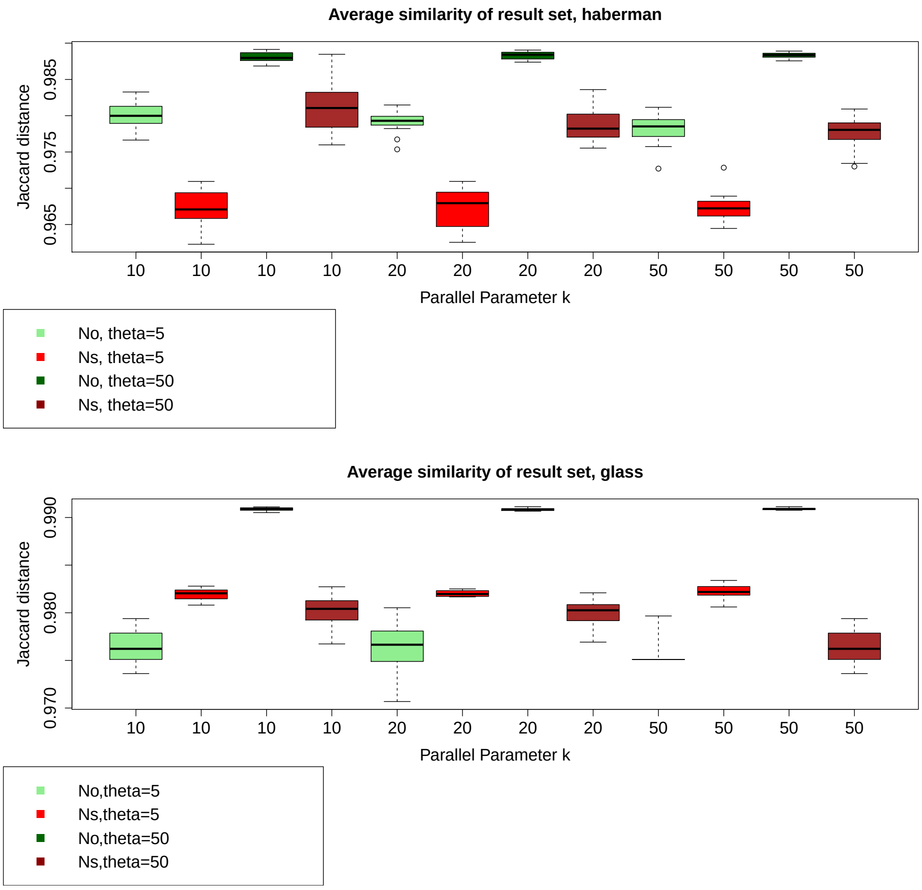


Figure 8.6: Average Jaccard distance of the models in the solution set obtained by Widening via similarity neighborhoods compared to the average Jaccard distance of the models in the solution set discovered by Widening via optimality neighborhoods for the glass and haberman data sets.

8.4.5 Similarity Search using Widening via Similarity Neighborhoods.

The second potential application of the Widening via similarity neighborhoods is the similarity search. For that we need to look at the average quality of the obtained set of k models at each run, as well as the average pairwise similarity of the obtained set of models at each run of the widened search. As can be seen in Figures 8.9, 8.8, 8.6, 8.7, the sets of models obtained by N_k^s perform well on average and have higher similarity than the models obtained by Widening via optimality neighborhoods (although for this neighborhood size, the difference is not big). It can be seen that the average

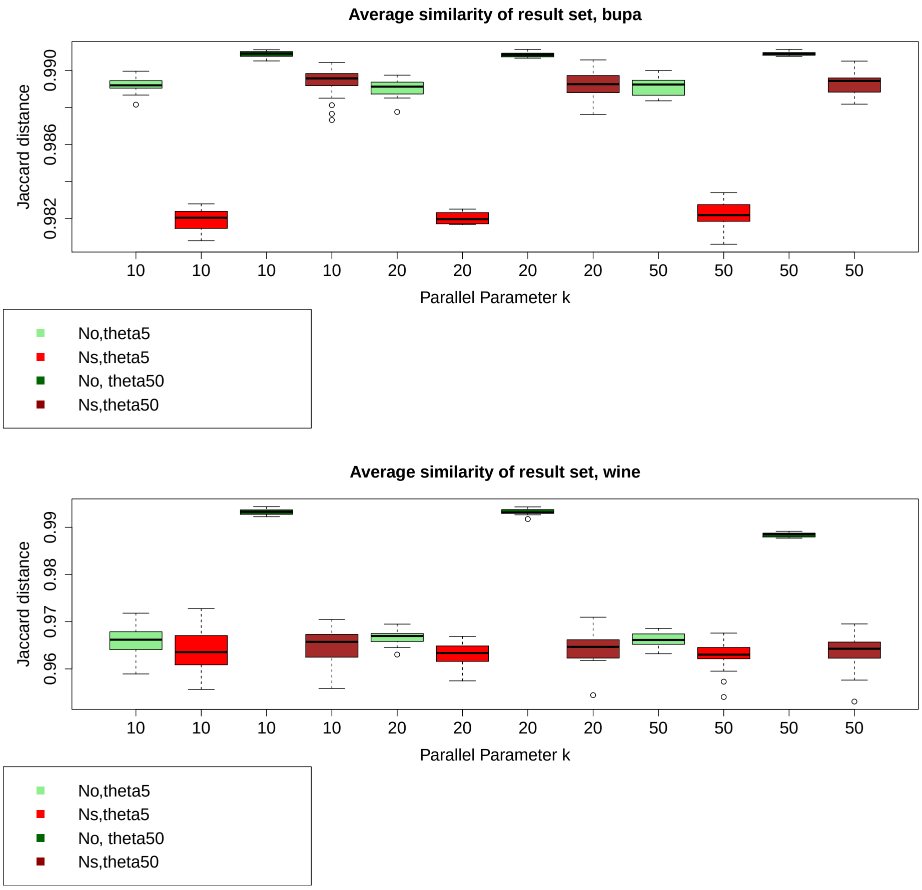


Figure 8.7: Average Jaccard distance of the models in the solution set obtained by Widening via similarity neighborhoods compared to the average Jaccard distance of the models in the solution set discovered by Widening via optimality neighborhoods for the bupa and wine data sets.

quality of the set of k discovered models is mostly equal to (or better than) greedy. The similarity of the models, discovered by Widening via similarity neighborhoods can be seen in Figures 8.6, 8.7, where they are contrasted with the similarity between the set of models discovered by Widening via optimality neighborhoods. Widening via similarity neighborhoods produces models that are more similar in comparison to the Widening via optimality neighborhoods. The similarity between the models is not drastically higher compared to that of the set discovered by Widening via optimality neighborhoods, but will be better with a greater number parallel workers. Furthermore, these results show that the Widening via neighborhoods is a sparse method, and in order to improve the similarity between the models in the resulting set of models, a smaller neighborhood

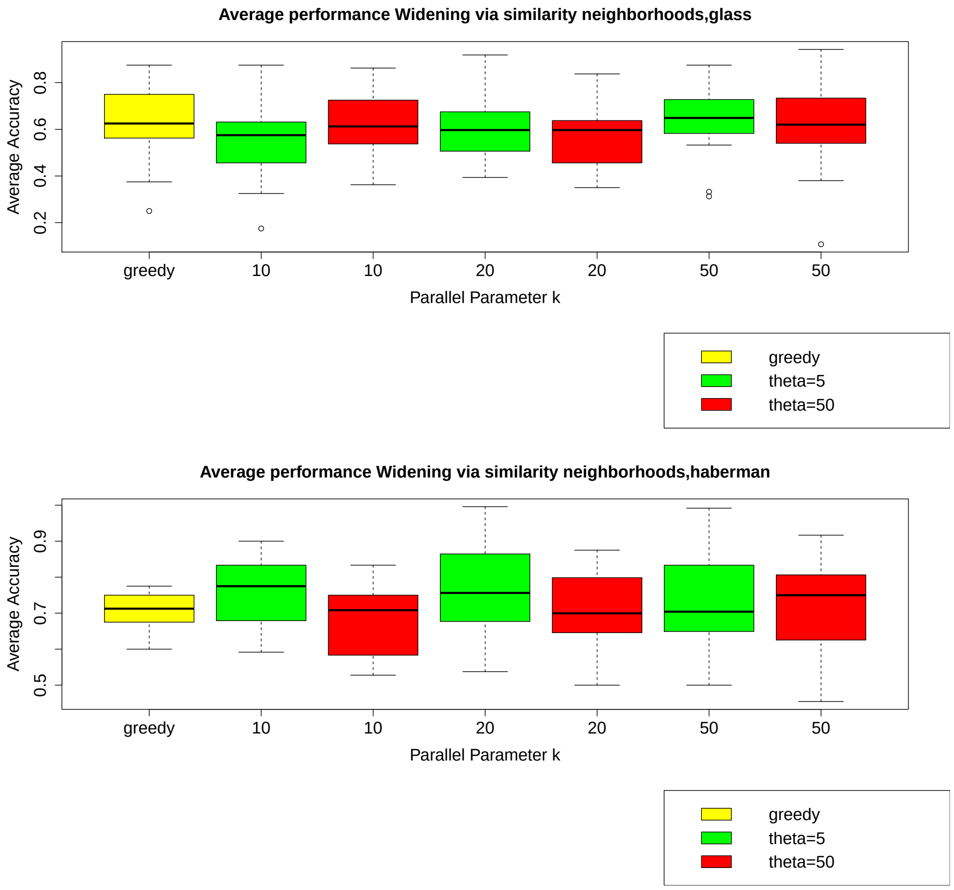


Figure 8.8: Average performance of the models in the solution set obtained by by Widening via similarity neighborhoods for the glass and haberman data sets.

size θ needs to be used. As can be seen in Figures 8.9, 8.8, 8.6, 8.7, the sets of models obtained by N_k^s perform well on average and have higher similarity than the models obtained by Widening via optimality neighborhoods (although for this neighborhood size, the difference is not big). It can be seen that the average quality of the set of k discovered models is mostly equal to (or better than) the greedy approach. The similarity of the models, discovered by Widening via similarity neighborhoods can be seen in Figures 8.6, 8.7, where they are contrasted with the similarity between the set of models discovered by Widening via optimality neighborhoods. Widening via similarity neighborhoods produces models that are more similar in comparison to the Widening via optimality neighborhoods, this is especially valid for a larger θ. The similarity between the models is not drastically higher compared to that of the set discovered by Widening via optimality neighborhoods, but will be better with a greater number parallel workers.

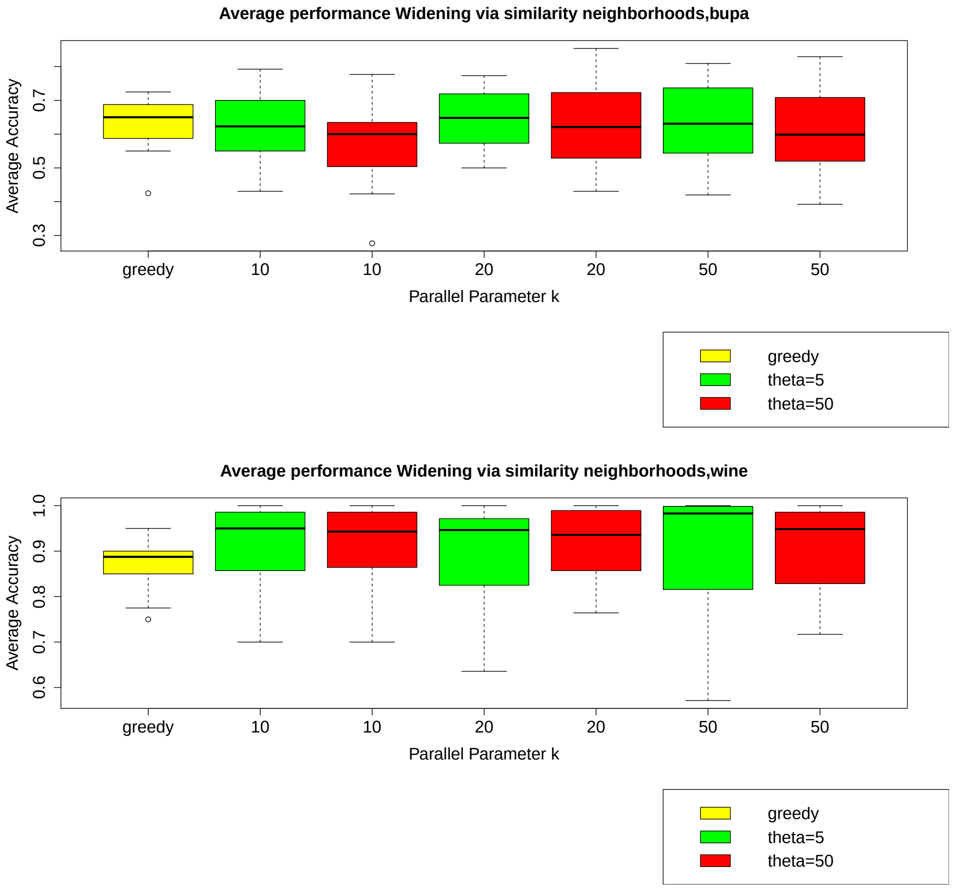


Figure 8.9: Average performance of the models in the solution set obtained by by Widening via similarity neighborhoods for the bupa and wine data sets.

Furthermore, these results show that the Widening via neighborhoods is a sparse method, and in order to improve the similarity between the models in the resulting set of models, a smaller neighborhood size θ needs to be used.

8.4.6 Widening via Diverse Neighborhoods

The results presented in Figures 8.14, 8.15 show the performance of Widening via diverse neighborhoods for different values of the diversity threshold δ. Limiting the size of the diverse neighborhood, θ, focuses the search on a few diverse and best performers at each step. The optimal value of the threshold will depend on the structure of the search space. Once again, the results demonstrate that diversity improves the search of the space by discovering diverse and good solutions, and the best solution discovered is

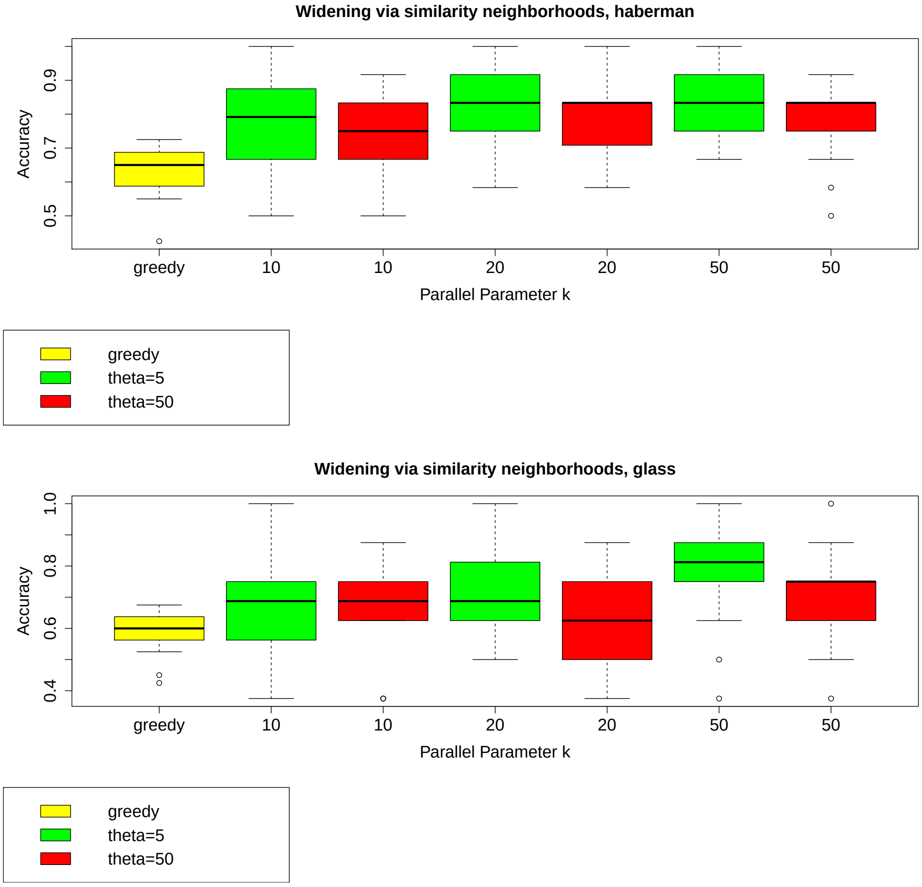


Figure 8.10: Performance of Widening via similarity neighborhoods for different values of parallel parameter k and neighborhood size θ for the haberman and glass data sets.

on average better. Different parameter tunings are required to achieve optimal results. The results obtained are comparable with diverse $Top - k$ approach. As can be seen in Figures 8.12, 8.13, where we contrast Widening via optimality neighborhoods with Widening via diverse neighborhoods, diversity can improve the results in comparison to the neighborhoods-based approach which does not use diversity. The results obtained by Widening via diverse neighborhoods are comparable to those obtained using $Top-k$, given the appropriate threshold. The size of the diverse neighborhood in these experiments is relatively small, $\theta = 5$. From the runtime experiments in Section 8.4.7, it is clear that the size of the neighborhood, and building the diverse neighborhoods dominate the runtime of the search. With a small enough neighborhood, the running time is not significantly different from constant, and is significantly better compared to $Top-k$ Widening. Depending on the data set and the threshold used, small neighborhoods can

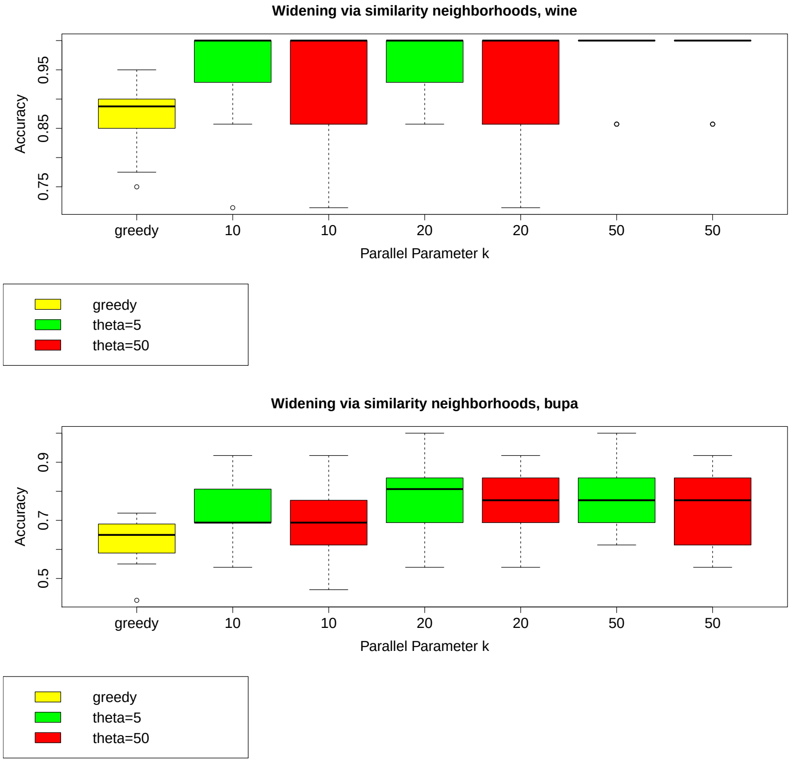


Figure 8.11: Performance of Widening via similarity neighborhoods for different values of parallel parameter k and neighborhood size θ for the bupa and haberman data sets.

already produce high quality results. The results show that adding diversity in a controlled manner can improve the quality of the final result. The relative merits of the different communication-less methods seem to be data-dependent. Using Widening via $N^{o}_{k,\theta}$ allows for more intensified versus more dispersed search, with giving higher priority to diversity or to optimality. Increasing the number of parallel workers for a fixed θ will allow for giving higher priority to optimality compared to diversity, by guiding the search to investigate the peaks with highest optimality. Increasing θ and keeping parameter k constant leads to more dispersed searches throughout the search space. We compare the performance of Widening via neighborhoods for different values of parameters k and θ. In all approaches of diversity-driven Widening, the main issue is the trade-off between model quality and diversity and the optimal balance between the two will depend on the landscape of the particular search space. Experimentally, the communication-less methods for diversity-driven Widening can compare with the ones using communication. Using larger number of parallel workers in a smart way can compensate for the lack of communication.

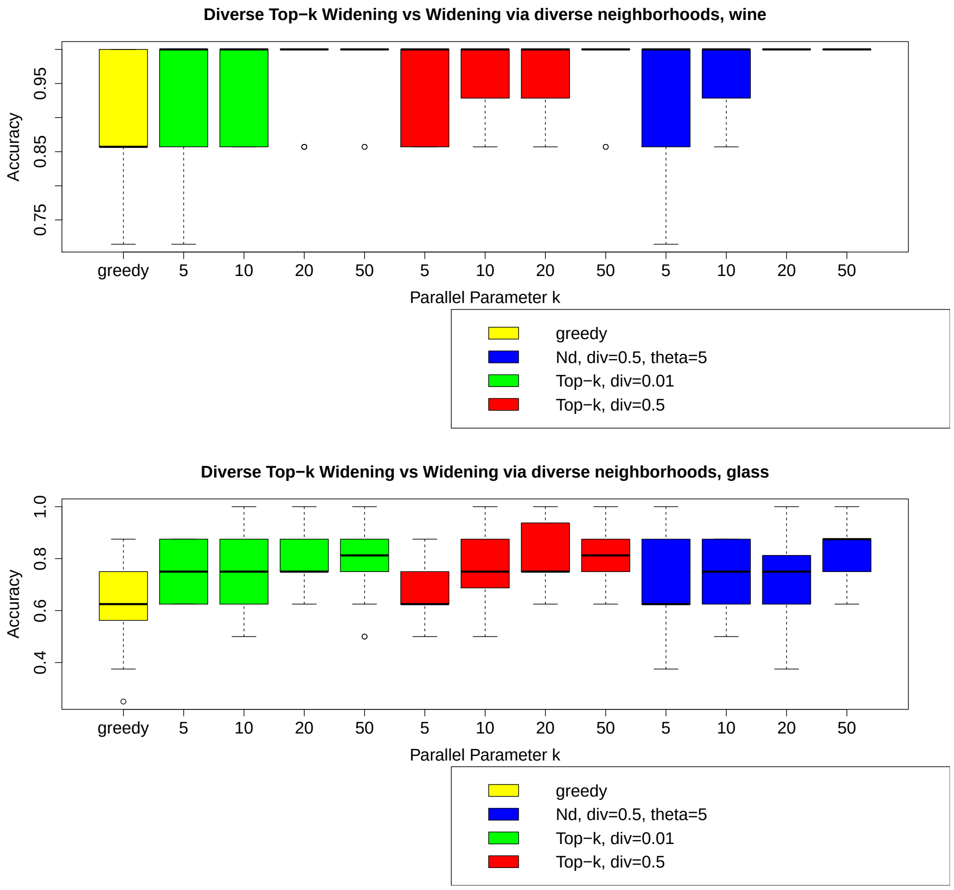


Figure 8.12: Widening via diverse neighborhoods, diversity obtained through a fixed threshold compared to diverse $Top - k$ Widening, the data sets are wine and glass.

8.4.7 Running Time Experimental Results

Analysis of the Experimental Results

For a small enough size of the neighborhood, communication-less methods have better running times than $Top-k$, it is clear that communication-less methods are significantly better than the approach which requires communication. The size of the neighborhoods is the biggest influence on the running time in Widening via neighborhoods approaches. The number of parallel workers does not influence the running time as much, given enough parallel resources.

Another factor which can influence the running time significantly, is the time spent on evaluating the similarity or distance between two models. This depends both on implementation and on the dimension of the search space. The higher the dimensions,

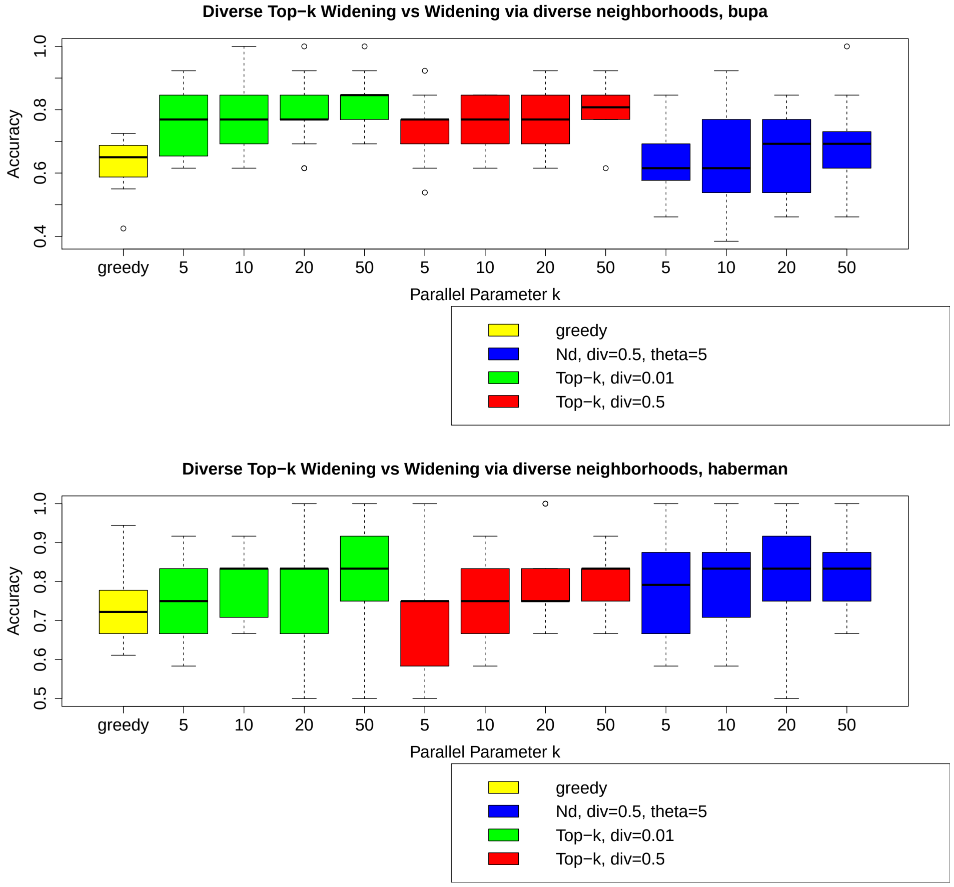


Figure 8.13: Widening via diverse neighborhoods, diversity obtained through a fixed threshold compared to diverse $Top - k$ Widening, the data sets are bupa and haberman.

the more coordinates need to be compared. A similarity measure which is costly has a big effect on the running time. The implementation can also greatly affect the results, using a strategy with a fast lookup is important. For the data-based diversity this is of special importance, because a look up needs to be performed for every data point. The Jaccard distance is used, which looks up every data point covered by a given rule. If this look up is implemented in a costly way calculation of the Jaccard distance is very expensive time-wise. The running time depends also on the complexity of the diversity measure. A diversity measure that is simple to calculate is not going to affect the running time as much. Another factor, which influences the running time of Widening via diverse neighborhoods, is the threshold used. The higher the threshold, the more comparisons need to be make, and the longer it takes to build the neighborhood. However, for the data sets used, the size of the neighborhood affects the running time more than the

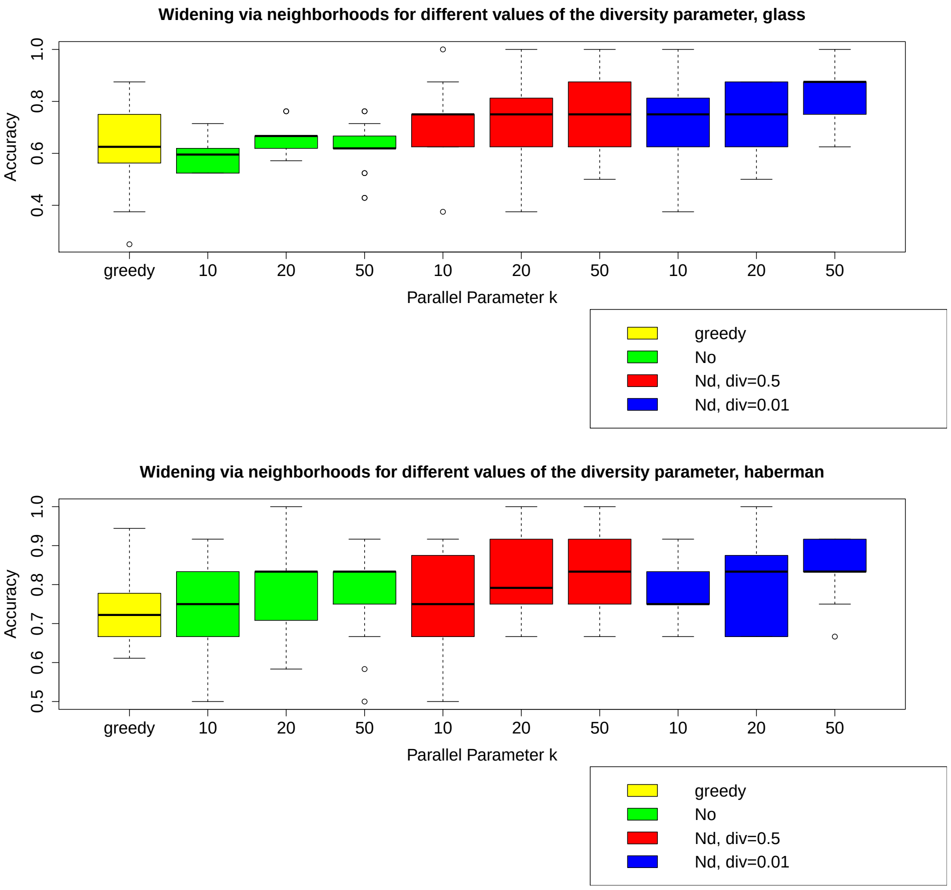


Figure 8.14: Widening via diverse neighborhoods, for different values of Widening parameter k and different values of the diversity threshold compared to diverse $Top - k$ Widening, the data sets are haberman and glass. The size of the neighborhood is $\theta = 5$.

diversity threshold used.

This is also compatible with the theoretical analysis of the running time, due to the fact that each worker in parallel has to build the neighborhood by itself. For small neighborhoods, Widening via diversity neighborhoods is better than Widening via similarity neighborhoods. Widening via similarity neighborhoods is less influenced by the size of neighborhood compared to the Widening via diversity neighborhoods, which is compatible with the theoretical analysis in Section 4.10.5. In similarity neighborhoods, the similarity of all refinements is evaluated and then the θ most similar are chosen, while in Widening via diversity neighborhoods using a threshold the refinement sets are searched only until θ diverse models, which satisfy the threshold δ, are found. Widening via similarity neighborhoods can benefit from preprocessing. The running time is also

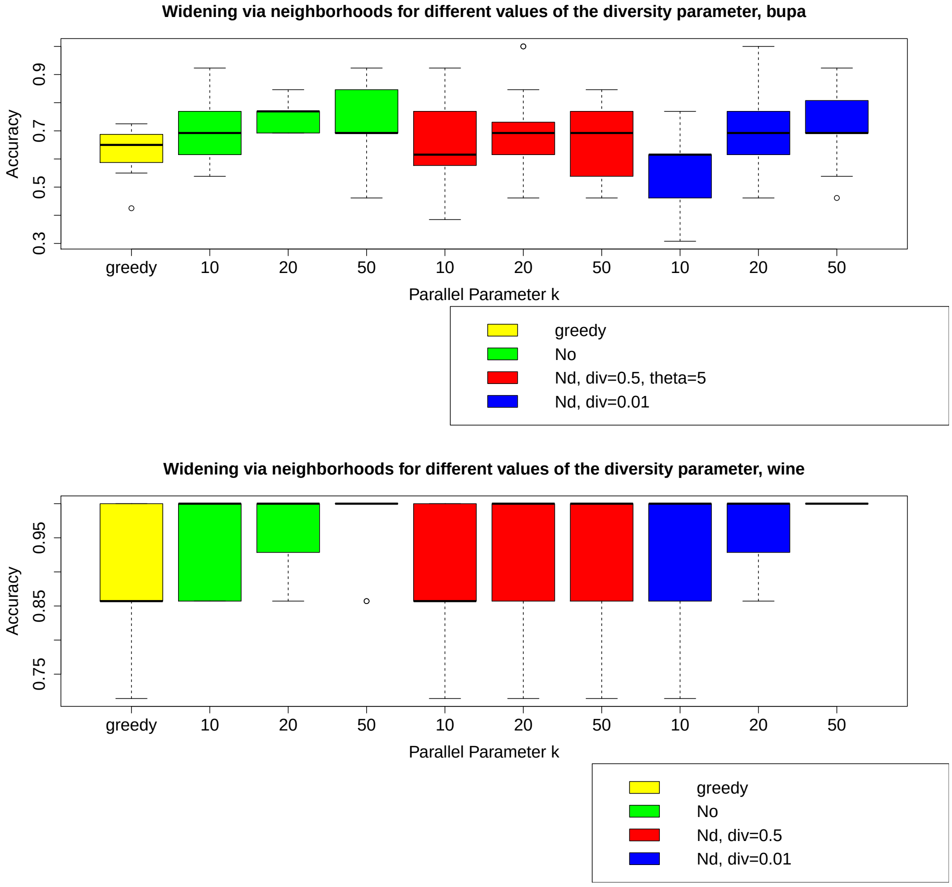


Figure 8.15: Widening via Diverse Neighborhoods, for different values of Widening parameter k and different values of the diversity threshold, the data sets are bupa and wine. The size of the neighborhood is $\theta = 5$.

affected by the type of similarity measure used and the implementation of calculation of similarity measure. Due to the high number of comparisons needed to select the top $k-1$ most similar neighbors the similarity evaluation needs to be implemented in a way which is time efficient. Efficient ways to perform nearest neighbor searches will improve the running time of the neighborhood-based methods. Preprocessing before the search can be used, because the model fragments are known before the search.

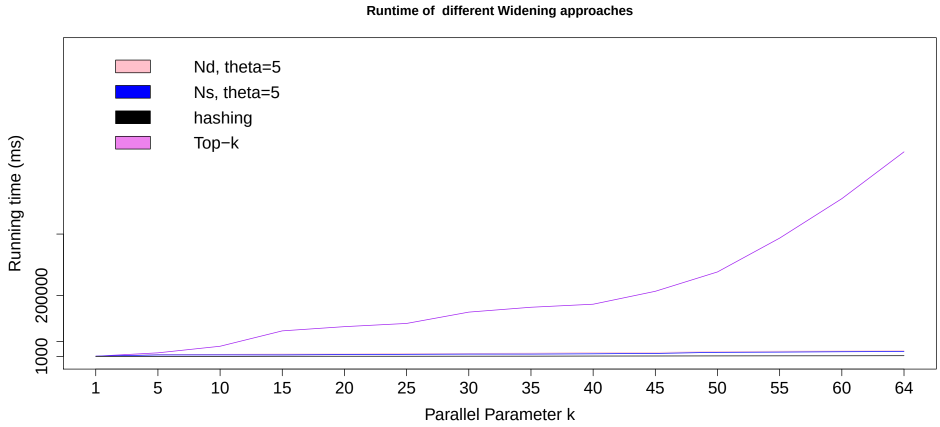


Figure 8.16: Comparison of the runtime for different Widening methods using the *haberman* data set. All the neighborhood-based approaches use a small neighborhood size $\theta = 5$.

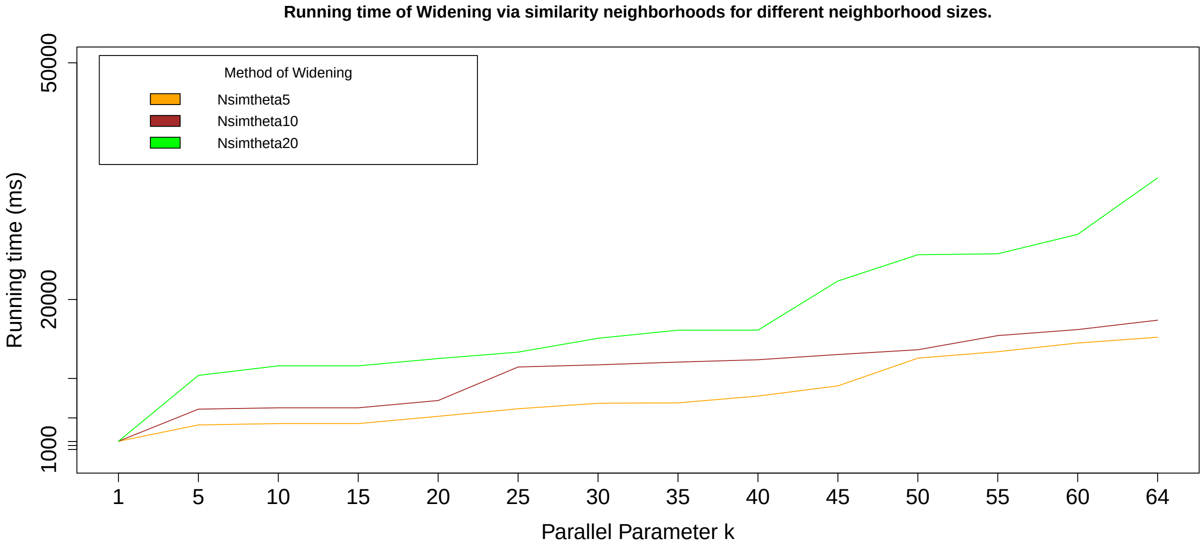


Figure 8.17: Comparison of the running times for Widening via similarity neighborhoods for different parameters using the *haberman* data set.

8.5 Conclusions

All Widening approaches show improvement of the solution quality, when compared with the greedy solution. Increasing the number of parallel workers improves the solution qual-

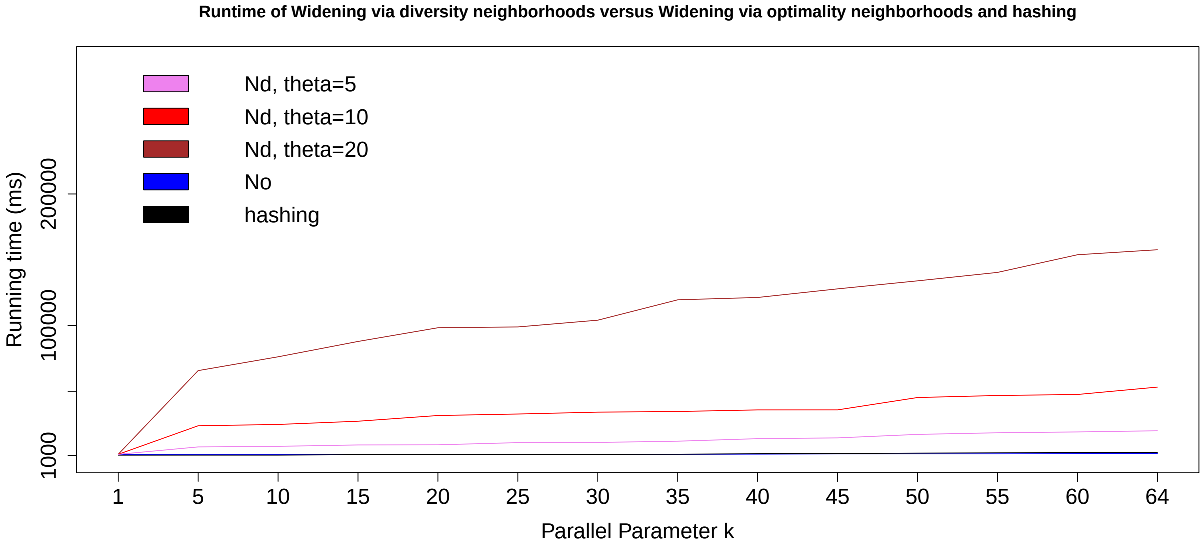


Figure 8.18: Comparison of the runtime of Widening via diverse neighborhoods with different neighborhood size versus Widening via optimality neighborhoods and Widening via hashing using the *haberman* data set.

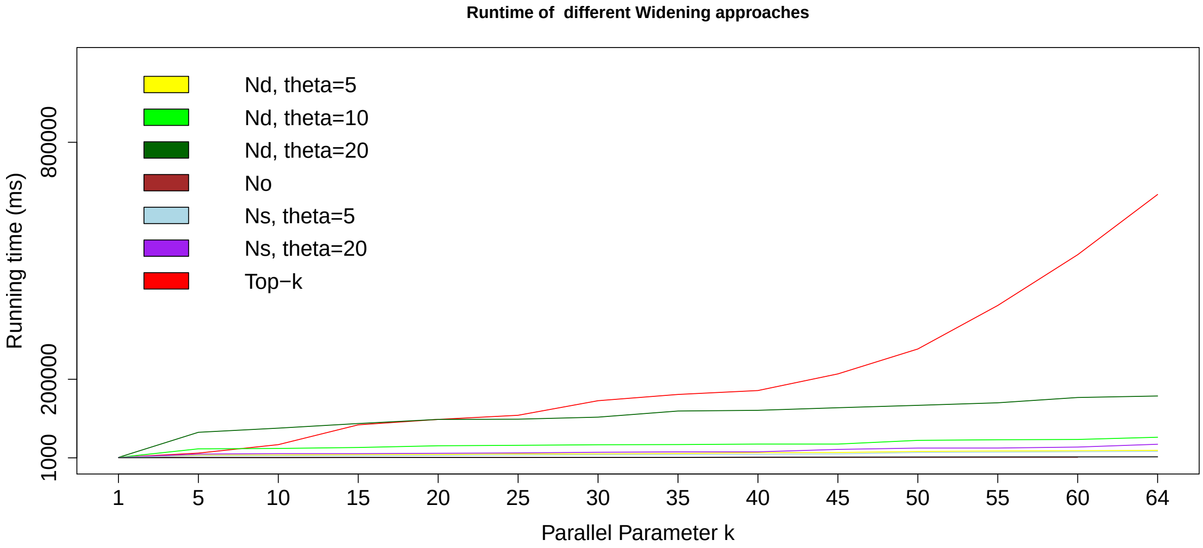


Figure 8.19: Comparison of the runtime for different Widening methods using the *haberman* data set.

ity. The appropriate use of diversity improves the solution quality. Communication-less Widening has the potential to compare with the $Top - k$ Widening, when done right. Additionally, adding more parallel workers can compensate for the lack of communication. A larger size of the neighborhood does not lead to improved results. However, diversity in combination with model quality, which leads to the investigation of promising solutions, provides an improvement of the quality of the final solution. Different approaches to diversity achieve different levels of diversity in the end result. Dependent on the type of the data and search space landscape, some diversity approaches are more beneficial than others. Widening via similarity neighborhoods is successful in discovering solutions with similar properties, and good quality. Using similarity neighborhoods in combination with diverse neighborhoods will help investigate small similarity neighborhoods of various promising solutions and have the chance to perform better than larger similarity neighborhoods. It is clear that the size of the neighborhood is decisive for the running time and dominates the widened search. This especially holds for Widening via diverse neighborhoods, where the size of the neighborhood dictates the amount of calculations needed to build the neighborhood. For Widening via similarity neighborhoods, the neighborhood size is not that influential, since the work is the same – the similarity is calculated for all the members of a refinement set, regardless of the size of neighborhood. The similarity neighborhoods are built using a comparison only to the locally optimal model, however, all members of the refinement set are compared to that refinement. If a similarity neighborhood is built using a fixed threshold, instead of the k most similar, then the running time can be improved. When it comes to Widening via similarity neighborhoods, it is most often more advantageous to use a small size of the neighborhood, since the goal is to find a set of very similar and good models. Even with regards to Widening via diverse neighborhoods, a larger neighborhood size is not always more advantageous than a smaller one, due to the fact that this method is very sparse. A large number of parallel workers and Widening with a smaller neighborhood size can be sufficient for discovering many promising solutions. The threshold δ does not influence the running time this much, so it is better to use the threshold in order to control the amount of diversity when building diverse neighborhoods, instead of aiming at large diverse neighborhoods. The lack of communication can be (partially) compensated by more work done in parallel. For very large size of neighborhoods, the running time can be equal or worse than that of $Top - k$ Widening, due to the need for a very large part of the landscape to be built by each individual worker, even though synchronization between the parallel workers is not necessary. We demonstrated (in Figures 8.12, 8.13) that even with a small size neighborhood θ, the obtained model quality can be comparable to the $Top - k$ Widening approach. While this result is data-dependent, in general increasing the diversity threshold can compensate for the size of the neighborhood.

Chapter 9

Outlook and Future Work

In this work, we investigate different Widening approaches with and without communication, both theoretically and experimentally. We implement said approaches to two widely used algorithms in data mining. The greatest challenge is to implement diverse structured search of the search space without using communication between the parallel workers. In order to compensate for the lack of communication, each parallel worker has to do more work to investigate a larger portion of the local landscape, or some knowledge of the search space must be encoded via pre-processing.

9.1 Future Work focused on Improving the Running Time of Neighborhood-based Widening

The bottleneck of Widening via neighborhoods is the calculating and building of the neighborhoods, and depends on the neighborhood size. This is especially true for the diverse neighborhoods approaches. First, different pre-processing techniques can be used in order to speed up the building of the neighborhoods. Many different methods for fast k nearest neighbors or approximate nearest neighbor searches exist. Such approaches include *local sensitivity hashing, $k-d$ trees*, and many others. These can be implemented on the space of model components, before the search starts and be used as pre-processing techniques with the goal of improving the running time. Then, instead of building a neighborhood at each step, the parallel worker can look up the needed neighbors, based on data-based similarity. For different types of situations, different techniques may be appropriate, for example, $k-d$ trees do not perform well on high-dimensional data.

Second, instead of pre-processing, different heuristics can be used to process large neighborhoods in a speedier manner. The most trivial technique to improve the running time is that instead of building the full neighborhood of size θ, each parallel worker builds the neighborhood of size at most that of its label (the size of neighborhood, which it needs). Because these labels are assigned at random, each parallel worker will build on average a neighborhood of size $\frac{\theta}{2}$ at each step. For more sophisticated approaches, techniques and heuristics developed for *very large neighborhood searches* can be used in order to gather more information about the larger portion of the landscape, while at the same time allowing fast processing. Additionally, smarter partitioning of each refinement set can be useful when improving the running time.

9.2 Future Work focused on Improving the Exploration of Neighborhood-based Widening

Similarity and diversity neighborhoods can be combined to improve the search. Diverse neighborhoods are used with the idea of exploration, while similarity neighborhoods are applied to achieve exploitation. Thus, in the beginning of the search diverse neighborhoods can be used to discover high quality solutions and later on similarity neighborhoods can help refine the search and discover better solutions.

9.3 Future Work Focused on the Algebraic and Topological Properties of the Refinement Graph used for Widening

In addition, for many problems the structure of the refinement graph of the search space can be known prior to the search. For different refinement operators the search space graph can have different properties. These properties can be used for thorough and full exploration of the search space, given enough parallel workers. For example, in this dissertation we used the fact that the Boolean lattice can be partitioned using chain decomposition, this is true for other types of posets. The knowledge of the various structures of the refinement graph of the search space can be used to partition the search space. Once global partitioning is achieved, different probabilistic approaches in different partitions can be used.

The refinement operator r for different algorithm types defines different types of poset topology. For example, the refinement operator of type 1, discussed in this dissertation, defines a lattice on $\mathcal{M}$. The topology of the poset, defined by the particular refinement operator, has different properties, which will help us define partitioning/traversals in parallel without the need of communication between the parallel workers. This topology and its properties can be calculated before the search.

For lattices many traversal algorithms already exist, some are also focused on efficiency. What is interesting as a future work is to use the existing traversal algorithms and apply them in parallel with the goal of Widening in mind – obtaining high quality solutions and, given enough parallel resources, ultimately discovering the optimal solution. Similarly, for other types of posets, algorithms may already exist that can be applied to the goal of Widening.

Another enhancement of the Widening approaches, which are based on global knowledge, is related to re-defining a redundant-free refinement operator. An additional improvement would be to define redundant free operators, which not only count identical model fragments as redundancies, but also very similar model fragments. Such synonyms can be defined using data-based similarity, and redundancy can be avoided by forbidding more than one synonym to be considered.

9.4 Future Investigations on the Basis of This Dissertation

Further improvements that I would like to achieve are as follows. The approaches will be tested on more data sets in the future. More data sets with different properties can provide an insight of how different topologies of the data can influence the relationship between the nodes in the refinement graph and perhaps help with enhancing Widening.

Moreover, the theoretical properties of the neighborhood-based Widening were investigated under very strict assumptions about the search spaces. These assumptions can be expanded. The theoretical properties of the Widening via diversity neighborhoods are a much more difficult problem, due to the fact that the whole search space landscape needs to be considered. Ideally, a specific performance-related measure needs to be evaluated: how many parallel workers are needed in order for the results from Widening via diverse neighborhoods to be similar to those of Widening via diverse $Top-k$.

If there is more knowledge about the relationship between the topology of the space of model components or the data topology and the search space of models, this prior knowledge will help to choose appropriate parameters and approaches for Widening.

9.5 Improving Widening via Neighborhoods by "Learning to Learn". Reinforcement Learning and Reactive Search

In this dissertation, we had a static approach to the neighborhoods themselves, we did not use any information from the structure of the data, the structure of the space of model components, nor from the subspaces of the refinement sets. A way to improve Widening is to use a reactive search approach, where the structure of a subspace of the search space "teaches" the parallel workers how to search. For example, parameters, determining search properties, can be optimized during the search, depending on the landscape. This can be done individually, as well as using a memory agent. Using a memory agent will require communication between reach individual parallel worker, but directly with it and if it is asynchronous it will not penalize the runtime too much.

Using information of the distribution of the model fragments with respect to similarity can also help choose appropriate thresholds for the diversity/similarity searches, proper neighborhood sizes and others.

Bibliography

[1] D.J. Newman A. Asuncion. UCI machine learning repository, 2007.

[2] Rakesh Agrawal and John C Shafer. Parallel mining of association rules. *IEEE Transactions on knowledge and Data Engineering*, 8(6):962–969, 1996.

[3] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In *VLDB*, volume 20, pages 487–499, 1994.

[4] Ravindra K Ahuja, Özlem Ergun, James B Orlin, and Abraham P Punnen. A survey of very large-scale neighborhood search techniques. *Discrete Applied Mathematics*, 123(1-3):75–102, 2002.

[5] R. M. Aiex, S. Binato, and M. G. C. Resende. Parallel grasp with path-relinking for job shop scheduling. *Parallel Comput.*, 29(4):393–430, April 2003.

[6] Renata M. Aiex, Mauricio G. C. Resende, Panos M. Pardalos, and Gerardo Toraldo. Grasp with path relinking for three-index assignment. *INFORMS Journal on Computing*, 17(2):224–247, 2005.

[7] Zaenal Akbar, Violeta N. Ivanova, and Michael R. Berthold. Parallel data mining revisited. Better, not faster. In *IDA*, 2012.

[8] Selim G. Akl. Parallel real-time computation: Sometimes quantity means quality. In *Computing and Informatics*, volume 21, pages 455–487. 2002.

[9] Enrique Alba. *Parallel Metaheuristics: A New Class of Algorithms.* Wiley-Interscience, 2005.

[10] Enrique Alba, Francisco Almeida, Maria J. Blesa, J. Cabeza, Carlos Cotta, M. Díaz, I. Dorta, Joachim Gabarró, C. León, J. Luna, Luz Marina Moreno, C. Pablos, Jordi Petit, A. Rojas, and Fatos Xhafa. Mallba: A library of skeletons for combinatorial optimisation (research note). In *Proceedings of the 8th International Euro-Par Conference on Parallel Processing*, Euro-Par '02, pages 927–932, London, UK, 2002. Springer-Verlag.

[11] Philippe Badeau, François Guertin, Michel Gendreau, Jean-Yves Potvin, and Eric Taillard. A parallel tabu search heuristic for the vehicle routing problem with time windows. *Transportation Research Part C: Emerging Technologies*, 5(2):109 – 122, 1997. Parallel Computing in Transport Research.

[12] S. Bailey, R. Grossman, H. Sivakumar, and A. Turinsky. Papyrus: A system for data mining over local and wide area clusters and super-clusters. In *Proceedings of the 1999 ACM/IEEE Conference on Supercomputing*, SC '99, New York, NY, USA, 1999. ACM.

[13] Roberto Battiti, Mauro Brunato, and Franco Mascia. *Reactive Search and Intelligent Optimization.* Springer Publishing Company, Incorporated, 1st edition, 2008.

[14] Yoshua Bengio et al. Learning deep architectures for ai. *Foundations and trends in Machine Learning*, 2(1):1–127, 2009.

[15] Michael R. Berthold, Nicolas Cebron, Fabian Dill, Thomas R. Gabriel, Tobias Kötter, Thorsten Meinl, Peter Ohl, Christoph Sieb, Kilian Thiel, and Bernd Wiswedel. Knime: The Konstanz information miner. In Christine Preisach, Hans Burkhardt, Lars Schmidt-Thieme, and Reinhold Decker, editors, *Data Analysis, Machine Learning and Applications*, pages 319–326, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[16] Michael R. Berthold, Oliver Sampson, Violeta Ivanova, and Arno Siebes. Widened data mining using parallel resources to improve model quality. 2013. unpublished manuscript.

[17] M. J. Blesa, L. Hernandez, and F. Xhafa. Parallel skeletons for tabu search method. In *Proceedings. Eighth International Conference on Parallel and Distributed Systems. ICPADS 2001*, pages 23–28, 2001.

[18] Leo Breiman. Bagging predictors. *JML*, 24(2), 1996.

[19] Leo Breiman. Random forests. *JML*, 45(1), 2001.

[20] Peter Bubenik. Statistical topological data analysis using persistence landscapes. *J. Mach. Learn. Res.*, 16(1):77–102, January 2015.

[21] Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.

[22] Gunnar Carlsson, Tigran Ishkhanov, Vin de Silva, and Afra Zomorodian. On the local behavior of spaces of natural images. *International Journal of Computer Vision*, 76(1):1–12, 2008.

[23] Gunnar Carlsson, Afra Zomorodian, Anne Collins, and Leonidas J Guibas. Persistence barcodes for shapes. *International Journal of Shape Modeling*, 11(02):149–187, 2005.

[24] I Chao, Bruce L Golden, Edward Wasil, et al. An improved heuristic for the period vehicle routing problem. *Networks*, 26(1):25–44, 1995.

[25] R. Chen, K. Sivakumar, and H. Kargupta. An approach to online bayesian learning from multiple data streams. In *In Proceedings of Workshop on Mobile and Distributed Data Mining, PKDD*, pages 31–45, 2001.

[26] Cheng-Tao Chu, Sang Kyun Kim, Yi-An Lin, YuanYuan Yu, Gary R. Bradski, Andrew Y. Ng, and Kunle Olukotun. Map-reduce for machine learning on multicore. In *NIPS*, 2006.

[27] Peter Clark and Robin Boswell. Rule induction with cn2: Some recent improvements. In *European Working Session on Learning*, pages 151–163. Springer, 1991.

[28] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1):103–120, 2007.

[29] Graham Cormode, Howard Karloff, and Anthony Wirth. Set cover algorithms for very large datasets. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 479–488. ACM, 2010.

[30] Teodor Gabriel Crainic, Gloria Cerasela Crisan, Michel Gendreau, Nadia Lahrichi, and Walter Rei. A concurrent evolutionary approach for rich combinatorial optimization. In *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*, GECCO '09, pages 2017–2022, New York, NY, USA, 2009. ACM.

[31] Teodor Gabriel Crainic, Beatrice Di Chiara, Maddalena Nonato, and Luciano Tarricone. Tackling electrosmog in completely configured 3G networks by parallel cooperative meta-heuristics. *IEEE Wireless Communications*, 13(6):34–41, 2006.

[32] Teodor Gabriel Crainic and Michel Gendreau. Cooperative parallel tabu search for capacitated network design. *Journal of Heuristics*, 8(6):601–627, 2002.

[33] Teodor Gabriel Crainic, Michel Gendreau, Pierre Hansen, and Nenad Mladenović. Cooperative parallel variable neighborhood search for the p-median. *Journal of Heuristics*, 10(3):293–314, 2004.

[34] Teodor Gabriel Crainic and Michel Toulouse. *Parallel Strategies for Meta-Heuristics*, pages 475–513. Springer US, Boston, MA, 2003.

[35] Teodor Gabriel Crainic and Michel Toulouse. Parallel strategies for meta-heuristics. In *Handbook of metaheuristics*, pages 475–513. Springer, 2003.

[36] George E Dahl, Dong Yu, Li Deng, and Alex Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on audio, speech, and language processing*, 20(1):30–42, 2012.

[37] Jianyong Dai, Joohan Lee, and Morgan C. Wang. Efficient parallel data mining for massive datasets: Parallel random forests classifier. In *PDPTA*, 2005.

[38] John Darlington, Yi-ke Guo, Janjao Sutiwaraphun, and Hing Wing To. Parallel induction algorithms for data mining. In *IDA*, 1997.

[39] Kenneth Alan De Jong. Analysis of the behavior of a class of genetic adaptive systems. *PhD thesis, University of Michigan*, 1975.

[40] Vin De Silva and Robert Ghrist. Coverage in sensor networks via persistent homology. *Algebraic & Geometric Topology*, 7(1):339–358, 2007.

[41] Jeffrey Dean and Sanjay Ghemawat. MapReduce: simplified data processing on large clusters. *Commun. ACM*, 51(1), 2008.

[42] Mary-Lee Dequeant, Sebastian Ahnert, Herbert Edelsbrunner, Thomas MA Fink, Earl F Glynn, Gaye Hattem, Andrzej Kudlicki, Yuriy Mileyko, Jason Morton, Arcady R Mushegian, et al. Comparison of pattern detection methods in microarray time series of the segmentation clock. *PLoS One*, 3(8):e2856, 2008.

[43] Tim Dettmers. 8-bit approximations for parallelism in deep learning. *arXiv preprint arXiv:1511.04561*, 2015.

[44] Inderjit Dhillon and Dharmendra Modha. A data-clustering algorithm on distributed memory multiprocessors. In *Large-scale Parallel KDD Systems Workshop, ACM SIGKDD*, 2000.

[45] Tapio Elomaa and Tuomo Malinen. On Lookahead Heuristics in Decision Tree. In *ISMIS*, pages 445–453, 2003.

[46] M. Elteir, H. Lin, W. c. Feng, and T. Scogland. Streammr: An optimized mapreduce framework for amd gpus. In *2011 IEEE 17th International Conference on Parallel and Distributed Systems*, pages 364–371, Dec 2011.

[47] Charles Epstein, Gunnar Carlsson, and Herbert Edelsbrunner. Topological data analysis. *Inverse Problems*, 27(12):120201, 2011.

[48] Saher Esmeir and Shaul Markovitch. Lookahead-based algorithms for anytime induction of decision trees. In *ICML*, pages 33–40, 2004.

[49] R. Farivar, A. Verma, E. M. Chan, and R. H. Campbell. Mithra: Multiple data independent tasks on a heterogeneous resource architecture. In *2009 IEEE International Conference on Cluster Computing and Workshops*, pages 1–10, Aug 2009.

[50] Ariel Felner, Sarit Kraus, and Richard E. Korf. KBFS: K-best-first search. *AMAI*, 2003.

[51] Thomas A Feo and Mauricio GC Resende. Greedy randomized adaptive search procedures. *Journal of global optimization*, 6(2):109–133, 1995.

[52] Charles Fleurent and Fred Glover. Improved constructive multistart strategies for the quadratic assignment problem using adaptive memory. *INFORMS Journal on Computing*, 11(2):198–204, 1999.

[53] Ian W. Flockhart and Nicholas J. Radcliffe. A genetic algorithm-based approach to data mining. In *KDD*, 1996.

[54] Michael J Flynn. Very high-speed computing systems. *Proceedings of the IEEE*, 54(12):1901–1909, 1966.

[55] Teodor Gabriel. *Parallel Meta-heuristic Search*, pages 1–39. Springer International Publishing, Cham, 2016.

[56] Félix García-López, Belén Melián-Batista, José A. Moreno-Pérez, and J. Marcos Moreno-Vega. The parallel variable neighborhood search for the p-median problem. *Journal of Heuristics*, 8(3):375–388, 2002.

[57] Ashwani Garg, Ashish Mangla, Neelima Gupta, and Vasudha Bhatnagar. PBIRCH: A scalable parallel clustering algorithm for incremental data. In *IDEAS*, 2006.

[58] Vijay K. Garg. *Introduction to Lattice Theory with Computer Science Applications*. Wiley Publishing, 1st edition, 2015.

[59] Michel Gendreau, François Guertin, Jean-Yves Potvin, and Éric Taillard. Parallel tabu search for real-time vehicle routing and dispatching. *Transportation Science*, 33(4):381–390, 1999.

[60] Michel Gendreau and Jean-Yves Potvin. *Handbook of metaheuristics*, volume 2. Springer, 2010.

[61] Robert Ghrist. Barcodes: the persistent topology of data. *Bulletin of the American Mathematical Society*, 45(1):61–75, 2008.

[62] Anglano Giordana, C. Anglano, A. Giordana, G. Lo Bello, and L. Saitta. A network genetic algorithm for concept learning. In *ICGA*, 1997.

[63] Attilio Giordana and Filippo Neri. Search-intensive concept induction. *Evolutionary Computation*, 3(4):375–419, 1995.

[64] Fred Glover. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 13(5):533 – 549, 1986.

[65] Fred Glover. Tabu search - part II. *INFORMS Journal on Computing*, 2(1):4–32, 1990.

[66] Fred Glover. Ejection chains, reference structures and alternating path methods for traveling salesman problems. *Discrete Applied Mathematics*, 65(1-3):223–253, 1996.

[67] Fred Glover and Manuel Laguna. *Tabu Search.* Kluwer Academic Publishers, Norwell, MA, USA, 1997.

[68] Fred Glover, Manuel Laguna, and Rafael Martí. Fundamentals of scatter search and path relinking. *Control and cybernetics*, 29(3):653–684, 2000.

[69] Fred Glover and Abraham P Punnen. The travelling salesman problem: new solvable cases and linkages with the development of approximation algorithms. *Journal of the Operational Research Society*, 48(5):502–510, 1997.

[70] Fred W Glover and Gary A Kochenberger. *Handbook of metaheuristics*, volume 57. Springer Science & Business Media, 2006.

[71] Diana Göhringer, Thomas Perschke, Michael Hübner, and Jürgen Becker. A taxonomy of reconfigurable single-/multiprocessor systems-on-chip. *International Journal of Reconfigurable Computing*, 2009, 2009.

[72] David E. Goldberg and Jon Richardson. Genetic algorithms with sharing for multimodal function optimization. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Application*, pages 41–49, Hillsdale, NJ, USA, 1987. L. Erlbaum Associates Inc.

[73] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* MIT Press, 2016.

[74] George Grätzer. *Lattice theory: foundation.* Springer Science & Business Media, 2011.

[75] Curtis Greene and Daniel Kleitman. Strong versions of sperner's theorem. *J. Comb. Theory, Ser. A*, 20:80–88, 01 1976.

[76] Chris Groër, Bruce Golden, and Edward Wasil. A parallel algorithm for the vehicle routing problem. *INFORMS J. on Computing*, 23(2):315–330, April 2011.

[77] Jerzy W Grzymala-Busse. Rule induction. In *Data mining and knowledge discovery handbook*, pages 277–294. Springer, 2005.

[78] Malay Haldar, Anshuman Nayak, Alok Choudhary, and Prith Banerjee. Parallel algorithms for fpga placement. In *Proceedings of the 10th Great Lakes Symposium on VLSI*, GLSVLSI '00, pages 86–94, New York, NY, USA, 2000. ACM.

[79] Eui-Hong Han, George Karypis, and Vipin Kumar. Scalable parallel data mining for association rules. In *SIGMOD*, volume 26, 1997.

[80] William D. Harvey and Matthew L. Ginsberg. Limited discrepancy search. In *IJCAI*, 1995.

[81] Bingsheng He, Wenbin Fang, Qiong Luo, Naga K. Govindaraju, and Tuyong Wang. Mars: A mapreduce framework on graphics processors. In *Proceedings of the 17th International Conference on Parallel Architectures and Compilation Techniques*, PACT '08, pages 260–269, New York, NY, USA, 2008. ACM.

[82] Giseon Heo, Jennifer Gamble, and Peter T Kim. Topological analysis of variance and the maxillary complex. *Journal of the American Statistical association*, 107(498):477–492, 2012.

[83] Daryl E. Hershberger and Hillol Kargupta. Distributed multivariate regression using wavelet-based collective data mining. *J. Parallel Distrib. Comput.*, 61(3):372–400, March 2001.

[84] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[85] Chuntao Hong, Dehao Chen, Wenguang Chen, Weimin Zheng, and Haibo Lin. Mapcg: Writing parallel program portable between cpu and gpu. In *Proceedings of the 19th International Conference on Parallel Architectures and Compilation Techniques*, PACT '10, pages 217–226, New York, NY, USA, 2010. ACM.

[86] Violeta Ivanova and Michael R. Berthold. Diversity-driven widening. In *Proceedings of the 12th International Symposium on Intelligent Data Analysis(IDA 2013)*, 2013.

[87] Violeta N. Ivanova-Rohling. Communication-less strategies for the widening of rule induction. In *Proceedings of the 19th International Conference on Computer Systems and Technologies*, CompSysTech'18, pages 33–37, New York, NY, USA, 2018. ACM.

[88] Violeta N. Ivanova-Rohling. Neighborhood-based strategies for widening of the greedy algorithm of the set cover problem. In *Proceedings of the 19th International Conference on Computer Systems and Technologies*, CompSysTech'18, pages 27–32, New York, NY, USA, 2018. ACM.

[89] Violeta N. Ivanova-Rohling. Properties of neighborhood-based approaches for widening. *Serdica Journal of Computing*, accepted for publication, 2018.

[90] Zongliang Jiang and Carla D. Savage. On the existence of symmetric chain decompositions in a quotient of the boolean lattice. *Discrete Mathematics*, 309(17):5278 – 5283, 2009.

[91] David S. Johnson. Approximation algorithms for combinatorial problems. In *STOC*, 1973.

[92] Dan Judd, Philip K. McKinley, and Anil K. Jain. Large-scale parallel data clustering. *TPAMI*, 20(8), 1998.

[93] Sanpawat Kantabutra and Alva L. Couch. Parallel k-means clustering algorithm on nows. *NecTec Technical Journal*, 2000.

[94] Hillol Kargupta, Byung-Hoon, Daryl Hershberger, and Erik Johnson. Collective data mining: A new perspective toward distributed data analysis. In *Advances in Distributed and Parallel Knowledge Discovery*, pages 133–184. AAAI/MIT Press, 1999.

[95] Hillol Kargupta, Joydeep Ghosh, Vipin Kumar, and Zoran Obradovic. Report from the workshop on distributed and parallel knowledge discovery. *SIGKDD Explor. Newsl.*, 2(2):108–109, December 2000.

[96] Hillol Kargupta, Weiyun Huang, Krishnamoorthy Sivakumar, and Erik Johnson. Distributed clustering using collective principal component analysis. *Knowledge and Information Systems*, 3(4):422–448, 2001.

[97] Hillol Kargupta, Weiyun Huang, Krishnamoorthy Sivakumar, Byung-Hoon Park, and Shuren Wang. *Collective Principal Component Analysis from Distributed, Heterogeneous Data*, pages 452–457. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000.

[98] Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[99] Charles E. Killian and Carla D. Savage. Venn diagrams and symmetric chain decompositions in the boolean lattice preliminary draft. 2002.

[100] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *SCIENCE*, 220(4598):671–680, 1983.

[101] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems*, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc.

[102] Richard Kufrin. Decision trees on parallel processors. In *PPAI*, 1995.

[103] Manuel Laguna, J. Wesley Barnes, and Fred Glover. Intelligent scheduling with tabu search: An application to jobs with linear delay penalties and sequence-dependent setup costs and times. *Appl. Intell.*, 3(2):159–172, 1993.

[104] Aleksandar Lazarevic and Zoran Obradovic. Boosting algorithms for parallel and distributed learning. *DPD*, 11(2):203–229, 2002.

[105] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[106] Soo-Young Lee and Kyung Geun Lee. Synchronous and asynchronous parallel simulated annealing with multiple markov chains. *IEEE Transactions on Parallel and Distributed Systems*, 7(10):993–1008, Oct 1996.

[107] Feiyue Li, Bruce Golden, and Edward Wasil. Very large-scale vehicle routing: new test problems, algorithms, and results. *Computers & Operations Research*, 32(5):1165–1179, 2005.

[108] Shen Lin and Brian W. Kernighan. An effective heuristic algorithm for the travelling-salesman problem. *Operations Research*, 21:498–516, 1973.

[109] Zhiqiang Ma and Lin Gu. The limitation of MapReduce: A probing case and a lightweight solution. In *Cloud Computing: Intl. Conf. on Cloud Computing, GRIDs, and Virtualization*, 2010.

[110] Samir W. Mahfoud. Crowding and preselection revisited. *Urbana*, 51:61801, 1992.

[111] Samir W. Mahfoud. *Niching Methods for Genetic Algorithms*. PhD thesis, University of Illinois at Urbana-Champaign, Champaign, IL, USA, 1995. UMI Order No. GAX95-43663.

[112] Samir W. Mahfoud and David E. Goldberg. Parallel recombinative simulated annealing: A genetic algorithm. *Parallel Comput.*, 21(1):1–28, January 1995.

[113] Simone L. Martins, Celso C. Ribeiro, and Mauricio C. Souza. *A parallel GRASP for the Steiner problem in graphs*, pages 285–297. Springer, Berlin, Heidelberg, 1998.

[114] Manish Mehta, Rakesh Agrawal, and Jorma Rissanen. Sliq: A fast scalable classifier for data mining. In *International Conference on Extending Database Technology*, pages 18–32, Berlin, Heidelberg, 1996. Springer.

[115] Sreerama Murthy and Steven Salzberg. Lookahead and pathology in decision tree induction. In *IJCAI(2)*, 1995.

[116] Quang Uy Nguyen, Xuan Hoai Nguyen, Michael O'Neill, and Alexandros Agapitos. An investigation of fitness sharing with semantic and syntactic distance metrics. In Alberto Moraglio, Sara Silva, Krzysztof Krawiec, Penousal Machado, and Carlos Cotta, editors, *Genetic Programming*, pages 109–120, Berlin, Heidelberg, 2012. Springer.

[117] Monica Nicolau, Arnold J Levine, and Gunnar Carlsson. Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival. *Proceedings of the National Academy of Sciences*, 108(17):7265–7270, 2011.

[118] Clark F. Olson. Parallel algorithms for hierarchical clustering. *JPC*, 21, 1995.

[119] Jong Soo Park, Ming-Syan Chen, and Philip S. Yu. An effective hash-based algorithm for mining association rules. *SIGMOD Rec.*, 24(2):175–186, May 1995.

[120] Jong Soo Park, Ming-Syan Chen, and Philip S. Yu. Efficient parallel data mining for association rules. In *Proceedings of the Fourth International Conference on Information and Knowledge Management*, CIKM '95, pages 31–36, New York, NY, USA, 1995. ACM.

[121] Stella C. S. Porto and Celso C. Ribeiro. Parallel tabu search message-passing synchronous strategies for task scheduling under precedence constraints. *Journal of Heuristics*, 1(2):207–223, 1996.

[122] R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2013.

[123] Thomas Rauber and Gudula Rünger. *Parallel programming: For multicore and cluster systems.* Springer Science & Business Media, 2013.

[124] U.K. Sarkar, P.P. Chakrabarti, S. Ghose, and S.C. Desarkar. Improving greedy algorithms by lookahead-search. *JAlgo*, 16(1), 1994.

[125] Robert E. Schapire. The strength of weak learnability. *JML*, 5, 1990.

[126] Jurgen Schmidhuber. Multi-column deep neural networks for image classification. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, pages 3642–3649, Washington, DC, USA, 2012. IEEE Computer Society.

[127] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[128] Jürgen Schulze and Torsten Fahle. A parallel algorithm for the vehicle routing problem with time window constraints. *Annals of Operations Research*, 86:585–607, 1997.

[129] John Shafer, Rakesh Agrawal, and Manish Mehta. Sprint: A scalable parallel classifier for data mining. In *VLDB*, 1996.

[130] Peter Shell, Juan Antonio Hernandez Rubio, and Gonzalo Quiroga Barro. Improving search through diversity. In *AAAI*, 1994.

[131] Takahiko Shintani and Masaru Kitsuregawa. Hash based parallel algorithms for mining association rules. In *DIS*, 1996.

[132] Anurag Srivastava, Eui-Hong Han, Vipin Kumar, and Vineet Singh. Parallel formulations of decision-tree classification algorithms. *DMKD*, 3(3), 1999.

[133] Salvatore J. Stolfo, Andreas L. Prodromidis, Shelley Tselepis, Wenke Lee, Dave W. Fan, and Philip K. Chan. JAM: Java agents for meta-learning over distributed databases. In *KDD*, 1997.

[134] Alexander Strehl, Gunjan K Gupta, and Joydeep Ghosh. Distance based clustering of association rules. In *Proceedings ANNIE*, volume 9, pages 759–764, 1999.

[135] Jeff A. Stuart and John D. Owens. Multi-GPU MapReduce on GPU clusters. In *Proceedings of the 2011 IEEE International Parallel & Distributed Processing Symposium*, IPDPS '11, pages 1068–1079, Washington, DC, USA, 2011. IEEE Computer Society.

[136] E.G. Talbi, Z. Hafidi, and J-M. Geib. A parallel adaptive tabu search approach. *Parallel Computing*, 24(14):2003 – 2019, 1998.

[137] Domenico Talia. Parallelism in knowledge discovery techniques. In *PARA*, 2002.

[138] Paul M. Thompson and Harilaos N. Psaraftis. Cyclic transfer algorithm for multivehicle routing and scheduling problems. *Operations Research*, 41(5):935–946, 1993.

[139] K. Tumer and J. Ghosh. Robust order statistics based ensembles for distributed data mining. In H. Kargupta and P. Chan, editors, *Advances in Distributed and Parallel Knowledge Discovery*, pages 185–210. AAAI/MIT Press, 2000.

[140] Sando Vega-Pons and Jose Ruiz-Shulcloprr. A survey of clustering ensemble algorithms. *IJPRAI*, 25(03), 2011.

[141] Marcus Gerardus Aldegonda Verhoeven and Emile HL Aarts. Parallel local search. *Journal of Heuristics*, 1(1):43–65, 1995.

[142] Abhishek Verma, Xavier Llorà, David E. Goldberg, and Roy H. Campbell. Scaling genetic algorithms using MapReduce. In *ISDA*, pages 13–18, 2009.

[143] Wikipedia contributors. Boolean algebra — Wikipedia, the free encyclopedia, 2018. [Online; accessed 30-July-2018].

[144] Wikipedia contributors. Multi-objective optimization — Wikipedia, the free encyclopedia, 2018. [Online; accessed 30-July-2018].

[145] David H. Wolpert. Original contribution: Stacked generalization. *Neural Netw.*, 5(2):241–259, February 1992.

[146] C Yu and DB Skillicorn. Parallelizing boosting and bagging. *Queen's University, Kingston, Canada, Tech. Rep*, 2001.

[147] M.J. Zaki. Parallel and distributed association mining: a survey. *Concurrency, IEEE*, 7(4), 1999.

[148] M.J. Zaki and C.T. Ho. *Large-Scale Parallel Data Mining*. Springer, 2000.

[149] Mohammed J Zaki. *Large-scale parallel data mining.* Number 1759. Springer Science & Business Media, 2000.

[150] Mohammed J Zaki. Parallel and distributed data mining: An introduction. In *Large-scale parallel data mining*, pages 1–23. Springer, 2000.

[151] Mohammed J. Zaki, Ching-Tien Ho, and Rakesh Agrawal. Parallel classification on SMP systems. In *IPPS*, 1998.

[152] Mohammed J. Zaki, Mitsunori Ogihara, Srinivasan Parthasarathy, and Wei Li. Parallel data mining for association rules on shared-memory multi-processors. In *SC*, page 43, 1996.

[153] Mohammed J. Zaki and Yi Pan. Introduction: Recent developments in parallel and distributed data mining. *DPD*, 11(2), 2002.

[154] Mohammed J. Zaki, Srinivasan Parthasarathy, Mitsunori Ogihara, and Wei Li. Parallel algorithms for discovery of association rules. *JDMKD*, 1(4):343–373, 1997.

[155] Weizhong Zhao, Huifang Ma, and Qing He. Parallel K-means clustering based on MapReduce. In *CloudCom*, pages 674–679, 2009.