Serdica J. Computing **14** (2020), No 3–4, 65–91

Serdica
Journal of Computing

Bulgarian Academy of Sciences
Institute of Mathematics and Informatics

THEORETICAL PROPERTIES OF A NEIGHBORHOOD-BASED APPROACH FOR WIDENING

Violeta N. Ivanova-Rohling

Abstract. Most of the research in parallel data mining and machine learning algorithms is focused on improving the efficiency of existing algorithms. However, our focus is the improvement of the solution quality, or model accuracy. We are looking for "smart" strategies to invest parallel compute resources in order to achieve a better exploration of the search space by exploring several solutions in parallel, referred to as *Widening*. In this paper, we discuss the theoretical properties of a neighborhood-based Widening using a type of neighborhoods, *optimality neighborhoods* and contrast this communicationless approach to the straightforward beam-like Top-k Widening approach, which requires communication. We show a bound on the number of parallel workers needed for the communicationless approach to guarantee that it has a solution of the same quality as the Top-k approach. In addition to the theoretical comparison, we experimentally compare these two approaches in terms of running time and quality of final solution, using a widened version of the greedy algorithm for set cover problem.

ACM Computing Classification System (1998): I.2.6, I.2.11, I.2.8.

Mathematics Subject Classification (2020): 68T20.

Key words: parallel neighborhood-based search, lattice traversal, parallel data mining, widening.

1. Introduction. The tendency towards pervasive and ever-increasing computing resources, such as GPUs, multicore processors, cloud computing and others, has led to intensive research in the area of parallel algorithms. Typically, parallel machine learning algorithms have been focused on improving the running time (improved efficiency) of the existing algorithms. However, parallel computing resources can be invested with the goal of improving the accuracy of the resulting model, or the quality of solutions obtained, instead of simply obtaining the same result, only faster. For many tasks, the quality of the obtained solution is of vital importance. This paper is focused on strategies for employing parallel computing resources in ways which improve the exploration of the search space and thus result in a better solution than a given heuristic. *Widening* is an umbrella term for strategies which use parallel simultaneous searches in the space of solutions in order to improve the result of a greedy heuristics. A simple Widening approach, which uses communication between the workers, was described in [17] and showed that an increase of parallel computing resources improves the quality of the discovered solution of the set cover problem. In addition to improving the quality of the solution, Widening aims at keeping the running time of the widened heuristic constant with respect to the number of parallel resources invested, and equal to that of the original greedy heuristic. In order to avoid undesired overhead, which arises from the communication between parallel workers, we are interested in strategies in which the parallel workers do not communicate when selecting paths through the search space. Communicationless Widening was already discussed in [17], where a simple approach using preferences was used. So far, the publications which discuss Widening strategies for the improvement of the quality of the solution are focused on experimental demonstration of the performance of the Widening methods for a large variety of algorithms. In contrast, in this paper we present an investigation of the *theoretical properties* of a new, local, neighborhood-based approach to Widening, which does not require communication. We estimate the number of parallel workers needed to have a result similar to that of the beam-like approach, called Top-k Widening, which uses communication. To do so, we use the known structure of the search space for a particular type of problems. We use our theoretical findings to determine which approaches can be used to further improve neighborhood-based Widening. We use experiments to illustrate the comparative characteristics of the two methods. We compare the running time of the Top-k Widening with that of the neighborhood-based method and contrast the running times of the two approaches with the quality of the solutions discovered by them.

Structure of This Paper. This paper is structured as follows. First, we present a general setting for Widening and define the main related concepts, define Widening as iterations of refinement and selection operators, motivate the need of diversity and the need for communicationless approaches. Then, we define communicationless Widening approach based on neighborhoods and discuss different types of neighborhoods. We proceed to define the refinement graph, and discuss the refinement graph structure for a special type of the refinement operator. We then investigate the performance of N_k^o. We are interested in how many parallel workers are needed so that the neighborhood-based approach can compete with the communication-based Top-k approach. We discuss first the case where the graph defined by the models which can be explored by N_k^o, G_{N_k}, has a uniform distribution of edges to each model at level l P^l and where P^l is strongly non-uniform. We discuss how the size of the neighborhoods affect the distribution of the paths P^l in G_{N_k}. We then proceed with practical experiments to compare how N_k and Top-k perform, as well as their running times. We conclude with discussing related work.

Goals of This Paper. We aim to compare theoretically as well as practically, using experiments, two Widening approaches: Top-k Widening and a communicationless approach, Widening via neighborhoods. We are interested in investigating how many parallel workers k are needed for the communicationless approach to perform as well as the approach which uses communication. We calculate different bounds including worst case bounds for the number of parallel workers needed. The motivation for this investigation is simple: communication between parallel workers has a strong effect on the running time of the widened algorithm.

2. General Widening of a Greedy Heuristic. Many data mining algorithms use greedy search strategies (or other types of heuristic approaches) through a space of potential solutions, choosing a locally optimal solution until a stopping criterion is satisfied. This heuristic search can 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 models. The *selection* operator chooses the locally best model from all possible refinements.

The selection operator is usually based on a given quality measure ψ, which evaluates the quality of a model m from a family of models $\mathcal{M}$: $\psi : \mathcal{M} \to \mathbb{R}$. According to this notation, one iterative step of the greedy search is represented as $m' = s_{\text{best}}(r(m))$, where

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

Namely, the model from the subset $M \subseteq \mathcal{M}$ which is ranked highest by the quality measure is chosen at each step. Figure 1 depicts the refinement/selection representation.

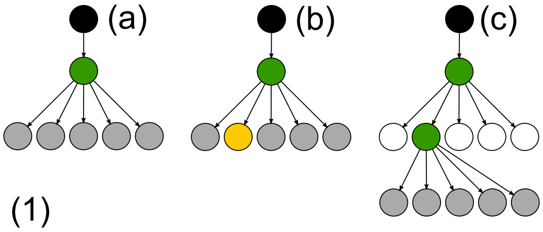

Fig. 1. The classic heuristic (often greedy) search algorithm. On the left (a), the current model m is depicted in green, the refinement options $r(m)$ are shown in gray. The selection operator s picks the yellow refinement (b) and the next level then continues the search based on this choice

It is known that unless the explored space is a matroid, choosing the locally optimal solution at every step will not guarantee discovering the global optimum. In order to improve the exploration of the search space, during Widening several solutions are chosen at each step, instead of the locally best one. One iteration of Widening can be represented as follows:

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

At each step, the *widened* selection operator s_w considers the refinements of a set M of original models and returns a new set M' of k refined models for further investigation. Parameter k is the *width* of the widened search. The larger the width, the higher the chances are of finding a better model in comparison to the normal greedy search. Figure 2 illustrates this process.

In [1] a beam-like approach, where the best k solutions are chosen at each step, was described and is referred to as Top-k Widening. 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 models are chosen:

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

where $s_{\text{Top-}k}$ selects the top k models from a set of models according to a given quality measure ψ. In [1] it was demonstrated that Top-k Widening leads to an

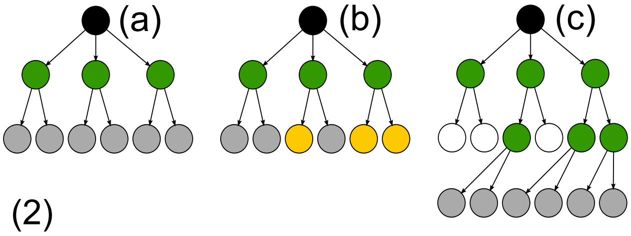

Fig. 2. Widening. From a set of models M (green circles), the refinement operator creates several sets of models (gray), shown on the left (a). The selection now picks a subset of the refined models (yellow circles in (b)) and the search continues from these on the right (c)

improved quality and that larger width leads to better accuracy. The main issue with Top-k Widening requires communication between the parallel workers, which has a strong influence on the running time. As stated already, Widening aims at preserving the running time the same as that of the greedy heuristic. Very simple communicationless strategies for Widening have already been described in [17]. The method, based on assignment of different orders of preferences, performs well in practice, but is not a structured predictable way to explore the search space.

In this paper we will investigate the theoretical properties of a communicationless neighborhood-based approach to Widening, called *Widening via optimality neighborhoods*, and estimate the number of parallel workers, needed to have a result similar to that of the Top-k Widening, which uses communication.

3. Communicationless Widening via Neighborhoods of Models. Ideally, we wish to split the search space among the parallel workers, so that each parallel worker explores a different partition. However, this is difficult due to the fact that initially we do not know the models in the search space. What we can do is use a localized approach: given a set of refinements M^r, we can directly force each individualized selection operator to pick a different refinement. That is, given a model m and its refinements $M^r = r(m)$, we assign the best k models to each of the k parallel workers. Below we formally define the concept of neighborhoods and describe a framework for this localized approach.

Definition 1. *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 $\forall i \in \{1, k-2\} : d(m'_i, m') < d(m'_{i+1}, m')$ and $\nexists m'' \in r(m) \setminus N_k(m') : d(m', m'') < d(m', m'_{k-1})$.*

In other words, a k-neighborhood of m', which is a refinement of model m, consists of the top k refinements of model m (including m'), which are closest to m' according to a distance d. Since we are interested in Widening of a greedy heuristic, which always selects a locally optimal choice, we define the neighborhoods with respect to the greedy choice. A schematic representation of k-neighborhood is seen in Figure 3.

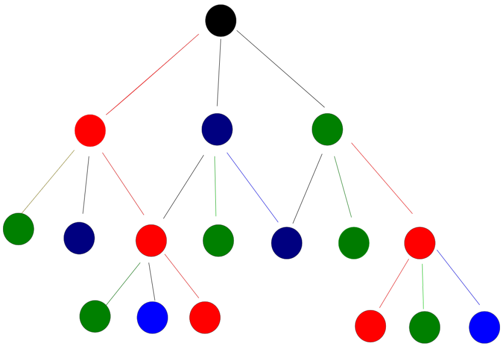

Fig. 3. Schematic representation for Widening via local partitioning of $M^r = r(m)$ using k-neighborhoods. Different neighbors are assigned to different parallel workers, represented by their color. Here $k = 4$. Each parallel worker chooses the model assigned to it *a priori*

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

Definition 2. *Given a selection operator s, a refinement operator r and a model m, let $m' = s(r(m))$. The set of selection operators $s^N_{Widening} = \{s_1^N, \ldots, s_k^N\}$ is a* ***Widening via k-neighborhoods*** *of s, if $\forall i \in \{1, \ldots, k\} : s_i^N(r(m)) = m'_i \in N_k(m') \wedge \forall i \neq j, i, j \in \{1, \ldots, k\} : m'_i \neq m'_j$.*

Different types of neighborhoods can be defined, by using a different distance measure d. Based on the type of neighborhood used, the Widening search will have different properties. For example, an **optimality k-neighborhood** is a k-neighborhood, where d is based on the original model evaluation function ψ. We will refer to Widening via optimality neighborhoods as N_k^o. Widening via optimality neighborhoods is an attempt to emulate the Top-k Widening in a communicationless 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. 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), where l is the refinement level. For very large k, N_k may stray too much away from the locally optimal solutions in a randomized fashion, to be useful. For small values of k, just like the Top-k search, it can converge to a local optimum.

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. In 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.

In this paper, we will discuss the properties of Widening via optimality neighborhoods.

Many different models are reachable via selection paths that share common initial subpaths, but then diverge, as shown in Figure 4. 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 3. *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 times).*

Namely, k models are selected from $N_\theta(s(r(m)))$. If $k \gg \theta$, this implies

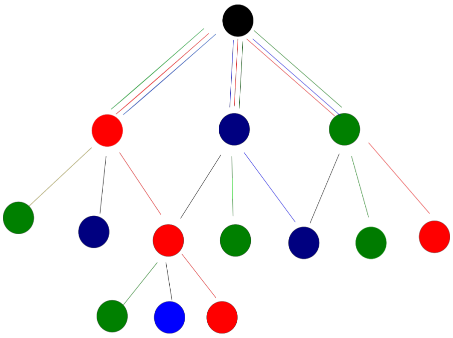

Fig. 4. Widening via θ, k-neighborhoods

repetitions between models $m'_0, m'_2, \ldots, m'_{k-1}$.

We will introduce a specific type of a k-neighborhood, *randomized k-neighborhood*, which we will use for our properties investigation.

Definition 4. *Given a family of models $\mathcal{M}$, we define* randomized k-neighborhood Widening*, Nr^o_k, as optimality k-neighborhood Widening, where each member of a given k-neighborhood is selected by a parallel worker with equal probability $\frac{1}{k}$.*

4. Search Space Graph $G_{\mathcal{M}}$. 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 $\forall m, m' \in \mathcal{M}, m' \in r(m) \exists e(m, m') \in E$. Clearly, $G_{\mathcal{M}}$ is a DAG.

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

Definition 5. *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.

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

Lemma 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 2. *Each node (model) at refinement level l is of size l and has l in-degrees.*

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

Lemma 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. □

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

Formal Definition of the SCP. We consider the standard (unweighted) set cover problem. Given a universe X of n items and a collection $\mathcal{S}$ of q subsets of $X : \mathcal{S} = \{S_1, S_2, \dots, S_q\}$. We assume that the union of all of the sets in $\mathcal{S}$ is X, with $|X| = n$: $\bigcup\limits_{S_i \in \mathcal{S}} S_i = X$. 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 $l = 0$ is the empty model $G^0 = \{m_0\}$, $m_0 = \{\}$. At level $l = 1$ the graph consists of each possible subset, provided by the problem. $G^1 = \{S_1, \dots, S_q\}$. 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 consists of the models of complexity l (i. e., models containing l subsets). The paths between nodes show the refinement relationship.

A Negative Example: Search Space Graph for Decision Trees

Definition 6. *We will call a refinement operator of type 2, r_2 if the model is not an unordered set of model fragments, where the refinement operator adds a new model fragment to this set, but is a collection of alternative unordered sets of model fragments.*

A refinement operator r for decision trees is not of type 1, but of type 2. Instead of a single unordered set of model fragments, a decision tree can be presented as a collection of alternative unordered sets of model fragments, which are alternative paths in the decision tree.

The refinement operator at a given step generates all possible refinements of model m by adding one model fragment, or attribute test, at one of the possible alternative unordered sets (branches).

Lemma 4. *The refinement graph of decision trees, built using a refinement operator r_2 as described in Definition 6 is not a lattice, but only a directed set.*

Proof. Consider two trees of depth 1, Imagine two decision trees of the type $a(b, c)$ and $c(d, e)$ which have no common model fragment. Then the following decision trees are both lower bounds according to the order defined by the refinement operator r_2, $a(b(d), c(e))$ and $c(d(a), e(b))$, however they do not have a greatest lower bound, because all the lower bounds are either incomparable to these two, or smaller, according to the order defined by r_2. □

5. Performance of N_k^o. The Widening approach N_k^o is a communicationless Widening strategy, which aims to explore the search space by considering not just the locally optimal choice, but also using the k optimality neighbors of the locally optimal choice in each refinement set $M^r = r(m)$. It aims to emulate in a communicationless way the Top-k Widening approach. By 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 selected from 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 communicationless Widening strategy can compete with the communication-heavy Top-k. 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 search space exploration.

Lemma 5. *Let $m_i, m_j \in \mathcal{M}$ be two distinct models, then the optimality neighborhoods 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 7. *Given a model evaluation function $\psi : \mathcal{M} \to \mathbb{R}$, and models $m_1, \ldots, m_k$ the function $s_{\mathit{FullTop}\text{-}k}$ is defined as follows:*

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

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 6. *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.

Definition 8. *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 $\frac{1}{k}$.*

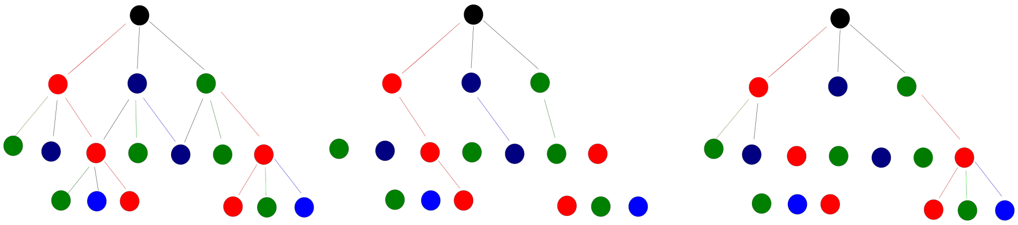

Fig. 5. The artificial FullTop-k structure, a bound for N_k^o and Top-k

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 9. *Given a set of models M and a model quality evaluation function $\psi : M \to \mathbb{R}$, we define a* performance-based distance $d_\psi : M \times M \to \mathbb{N}$ *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 $i = j$.*

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

Lemma 8. *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 9. *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 $Nr^o_k(\mathcal{M})$ 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^l_i for reaching a node m^l_i depends on the number of in-degrees to m^l_i 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 6: the probability of reaching the purple, blue or yellow node is two times greater than the probability of reaching the red node.

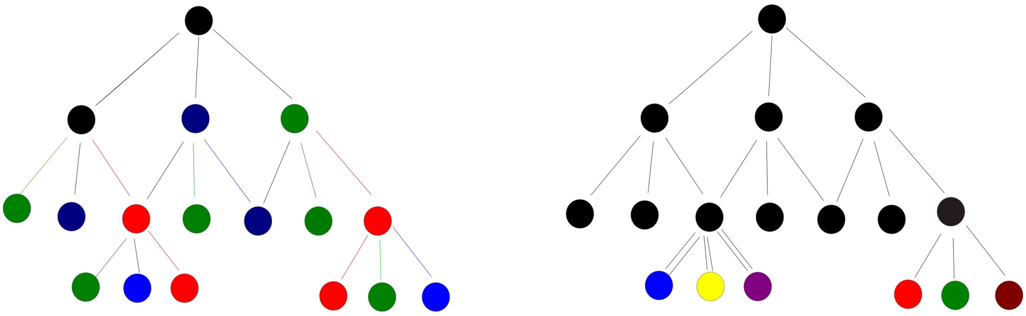

Fig. 6. A figure representing the Widening via randomized neighborhoods as random walks on the graph of models

Uniformly structured graphs lead to uniform P^l, probability distribution of reaching the nodes at level l. For very degenerate refinement graphs, where

P^l is strongly nonuniform, the set of solutions will consists of repetitions of some solutions and others may not be at all reached.

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 Section 5, 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.

Theorem 10. *Given that P^l is uniform, the solutions discovered by Widening via Nr^o_k, $\{Nr^o_k(\mathcal{M})\}$ $\in$$\{$FullTop-$k(\mathcal{M})\}$ are on the average uniformly distributed among the solutions $\{$FullTop-$k(\mathcal{M})\}$. Thus $\max\limits_{\psi}(\{(Nr^o_k)^l(\mathcal{M})\})$ will be on the average at most $\dfrac{k^{l-1}}{2}$ models away with respect to the model quality function ψ from $\max\limits_{\psi}(\{$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^o_k, since each path traversed by FullTop-k is equally likely to be traversed by Nr^o_k 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^k_o will be uniformly distributed among those k^l models of $\{$FullTop-$k^l(\mathcal{M})\}$. This implies that the $\max\limits_{\psi}(\{(Nr^o_k)^l(\mathcal{M})\})$ will be at most k^{l-1} models away from $\max\limits_{\psi}(\{$FullTop-$k^l(\mathcal{M})\})$. □

Upper Bound for the Number of Parallel Workers. We now want to derive an upper bound for the number of parallel workers k needed to guarantee $\{Nr^o_{\theta,k}\} = \{$Top-$k\}$. We will denote the performance of a given set of models M as Ψ. While it can be defined in various ways, such as the minimum, the maximum or the average solution quality of the models in the set, here we will define it as the maximum score among the models in the set. Namely,

$$\Psi = \max(\psi(m)), \forall m \in M$$

In the worst case, $max_{\psi}\{$Top-$\theta\} = max_{\psi}\{$FullTop-$\theta\}$. We want to have a quantitative estimation of 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

$\Psi(\{\text{Top-}\theta(\mathcal{M})^l\}) = \Psi(\{\text{FullTop-}\theta(\mathcal{M})^l\})$. In order to be able to guarantee that $\Psi(\{Nr^o_{\theta,k}(\mathcal{M})^l\})$ discovery of the best solution discovered by Top-θ, k needs to be large enough to discover every solution at level l. This is related to the number of paths in $G_{\text{FullTop-}\theta}$ at level l. In $G^l_{\text{FullTop-}\theta}$ the models at level l are at most θ^l.

Theorem 11. *We assume a uniform distribution P of the edges in $G_{FullTop\text{-}\theta}$. For $k = \min(\theta^l, \binom{n}{l})$, $Nr^o_{\theta,k}$ explores fully the models explored by FullTop-θ at step l and guarantees $\Psi(Nr^o_{\theta,k}(\mathcal{M}))^l \geq \Psi(\{Top\text{-}\theta(\mathcal{M})^l\})$.*

Proof. The graph $G_{\text{FT-}\theta}$ 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 $\Psi_{\max}\{N^o_{k,\theta}{}^l\} = \Psi_{\max}(\{\text{FullTop-}\theta^l\})$. □

In this worst case, in which Top-θ discovers the best model from $\{\text{FullTop-}\theta^l\}$, a significantly larger number of parallel resources are needed for the communicationless Widening approach to be able to guarantee the same performance as that of Top-k Widening.

$Nr^o_{\theta,k.}$ **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.

Theorem 12. *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 $k = \dfrac{1}{\min P^l(x)}$ parallel random walks each model at level l will be reached on average.*

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

In order to avoid this degenerate situation, there are several things to keep in mind. First, a small neighborhood size favors convergence to local optima, which is a big disadvantage especially for the communicationless method (although it is also a disadvantage of the Top-k, as it may explore too similar solutions in parallel). Furthermore, the use of diversity can help avoid degenerate graphs.

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 lattice 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 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 13. *Let $\{FullTop\text{-}k^l(\mathcal{M})\} = \{m_1^l, \ldots, m_p^l\}$, 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,

$$n_1, \ldots, n_p \to n.$$

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 $\max(\{\text{FullTop-}k^l(\mathcal{M})\})$.

Properties of θ, k-Neighborhoods. We can find how many parallel workers are needed for $\max\limits_{score}\{(N_{\theta,k}^o)^l(\mathcal{M})\}$ to be some distance from $\max\limits_{score}\{\text{Top-}\theta^l(\mathcal{M})\}$.

Theorem 14. *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-θ.*

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-θ. This follows from Lemma 10. $G_{\text{FT-}\theta}$ is the bound for Top-θ, and thus $max\{N_{\theta,k}^o\}$ is at most p models away in the graph $G_{\text{FT-}\theta}$ from $\max\limits_{\psi}\{\text{FullTop-}\theta\}$. □

6. Experimental Results and Discussion. In this section we will present experimental results from the application of Widening via optimality neighborhoods and Top-k Widening to the greedy algorithm for the set cover problem. We will compare the quality of the solution obtained (size of the cover) by both methods, and the effect that the number of parallel workers and size of neighborhood have on the quality of the solution. We will also compare the running time for different sizes of optimality neighborhoods to the running time of the Top-k Widening approach.

Greedy Set Covering. The greedy algorithm [18] 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. 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
    $S_{\text{current}} = \bigcup_{S \in C} S$
    $S_{\text{best}} = \arg\max_{S \in \mathcal{S}} \{|S \backslash S_{\text{current}}|\}$
    $C \leftarrow C \cup S_{\text{best}}$
until $\bigcup_{S \in C} S = X$;
return $C$.
```

Top-k Widening of the Greedy Algorithm for the Set Cover Problem. 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.

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.

Widening of the Greedy Algorithm for SCP via Optimality Neighborhoods. Each neighborhood is built on the refinement set $r(m)$ of a given model m. Let $m = \{S_i\}, i = 1, \ldots, l-1$. A refinement set $refine(m)$ consists of a set of models

$$\{\{S_i\} \cup S_{j_1}, \{S_i\} \cup S_{j_2}, \ldots, \{S_i\} \cup S_{j_{n-l+1}}\},\ i = 1, \ldots, l-1,\ S_{j_1}, \ldots, S_{j_{n-l+1}} \notin \{S_i\},$$

which differ in only one subset from each other, i. e., each of them contains $m = \{S_i\}, i = 1, \ldots, l-1$ and exactly one additional subset.

Then a k-neighborhood within the refinement set will contain k models, chosen from the refinement set of model m, which are chosen differently depending on the type of neighborhood. To each parallel worker one model from the k-neighborhood is assigned, with or without repetition.

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

Methods. All the approaches were implemented in Java, using KNIME [4]. Each experiment was run 50 times with shuffled order of the data.

Datasets. All the experiments are performed on three data sets rail507, rail516, rail582 from the OR Library database [3], which are associated with real-world set covering problems and have different properties. These data files arise from an application in Italian railways. The characteristics are as follows. The dataset rail507 has 507 rows and 63,009 columns, rail516 is with 516 rows and 47,311 columns, and rail582 is with 582 rows and 55,515 columns. As might be expected, these problems have a number of special characteristics, specifically: all column costs are either one or two, a column covers at most 12 rows.

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 communicationless methods.

Widening via Optimality θ, k-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.

Running Time Experiments. 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, from the OR library [3], `rail507` on a 64-core machine and repeated 10 times. Predictably, the number of parallel workers k had a strong influence on the running time of Top-k approach.

Implementation details in Java. The parallel search is implemented by a priority blocking queue. The parallel workers are independent threads, which are called from the thread pool and perform a search for solutions, based on a predefined behavior. For the Top-k approach, the threads are in communication with each other, by storing all their optimally discovered solutions in the priority blocking queue structure, from which the best k from the k^2 are then selected at each step and are used by the parallel workers as the next points of further exploration.

In the neighborhood-based approaches the parallel workers independently search through the space of potential solutions without exchanging information about the discovered solution. Prior to the search, each model fragment is assigned a neighbor list in the form of a hash table, which dictates which neighbor will be chosen by which worker, given that the model fragment is the optimal fragment to choose at a given step in the search. The result of the search of each thread (parallel search) is reported only at the very end of the search, and no intermediate solutions are communicated.

Results. In Figure 7 the effect of different neighborhood sizes is investigated. We can see that increasing θ while keeping the parallel workers constant can lead to worsening of the performance: a very large neighborhood size θ leads to a randomization of the search.

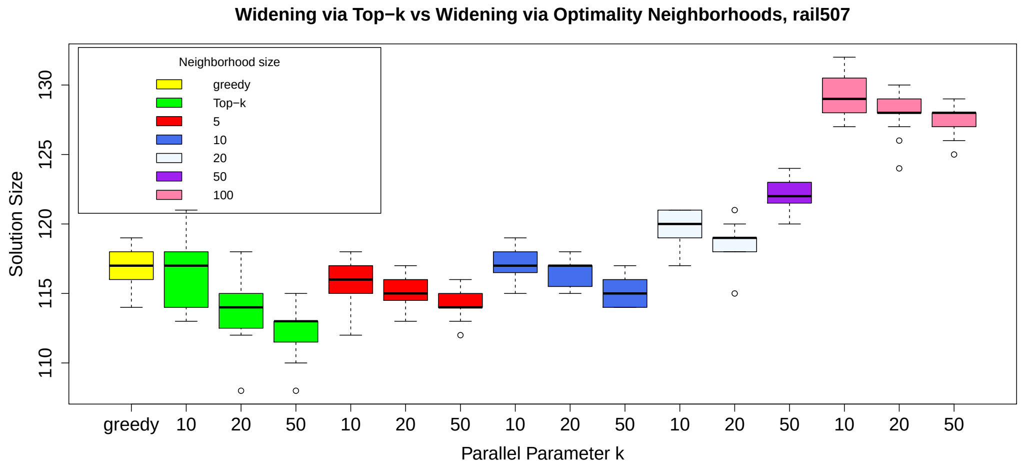

Fig. 7. Comparison of solution quality for Widening via optimality neighborhoods for different neighborhood size θ and Top-k Widening. Three data sets used: rail507, rail516, rail562 from the OR library [3]

As Figure 7 shows, the larger the number of parallel workers for a fixed neighborhood, the better the performance of the search. For a fixed number of parallel workers, increasing the size of the neighborhood eventually will lead to a randomized search. On the other hand, a small size of the neighborhood leads to exploring solutions, which are similar. While Widening via Top-k performs better than Widening via optimality neighborhoods, the performances of the two methods do not differ significantly. This may also be due to the fact that already the greedy algorithm for the set cover problem performs well. Further improvements on the results of Widening via optimality neighborhoods can be achieved via increased parallel resources or by explicit use diversity. This, however is not in the scope of this paper, which deals with investigating the theoretical properties of Widening via optimality neighborhoods.

Running Time of Optimality Neighborhoods. In this subsection we compare the experimentally obtained running times for both approaches using dataset *rail*507 `http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/rail507.txt`, from the OR library [3].

As expected, in Figure 8 one can see that the running time increases as the number of parallel workers increase. The experimental results confirm that the running time of the Top-k approach is significantly greater than the running time of Widening via optimality neighborhoods, while the model quality discovered by the two algorithms is similar. The communication between the parallel workers seems to have a big influence on the running time in the case of Top-k while this is not compensated by a great solution quality.

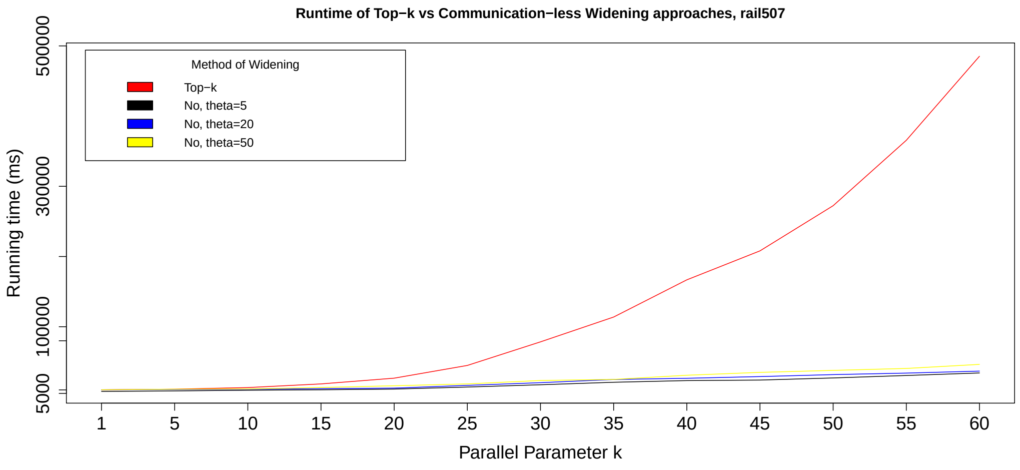

Fig. 8. Running time of Widening via optimality neighborhoods with different neighborhood size θ compared to the running time of Top-k

7. Evaluation and Summary of Obtained Results. We show that both Top-k and $\{N^{o}_{\theta,k}\}$ can be bound by G_{FT-k}. The refinement graph for the set cover problem is a lattice, and for large enough k the full refinement graph will be explored. The behavior of Widening via optimality neighborhoods depends a lot on the intersections between different neighborhoods, and distribution of edges in the graph G_{FT-k}. For large enough neighborhoods, however, the distribution of paths to each model at level l will be close to uniform, due to the fact that the refinement graph is a lattice. The size of the neighborhood has to be significantly smaller than the number of parallel workers, in case of a very non-uniform distribution, in order to be able to expect good performance from Widening via optimality neighborhoods. Experimental results show that while theoretically in general we cannot expect in the worst case, that Widening via neighborhoods performs as well as the Top-k, in practice Widening via neighborhoods performs comparatively well to the method, which uses communication, while at the same time having a much better running time.

8. Related Work

Speed-Up Through Parallelization. For the vast majority of parallelizations of data mining algorithms, the aim is to improve efficiency. Comprehensive surveys are found in [20,23,33,34]. A large amount of work focuses on the parallelization of decision tree learning. One of the earliest distributed decision tree algorithms, SPRINT [27], has served as the basis for many subsequent parallel decision tree approaches. Some noteworthy examples include [32] (employing data parallelism), [8] (using task parallelism), and [22, 28] (presenting hybrid approaches). Extensive surveys exist in the area of parallel association rule mining algorithms [31]. Parallelism in clustering algorithms has been used for both efficient cluster discovery and more efficient distance computations. Partitioning clustering algorithms are parallelized mostly using message-passing models, examples are presented in [9,19]. Examples for hierarchical clustering, which is more costly, include [11,25]. However, all these algorithms are using parallel techniques to achieve speed-up and not solution quality.

Model Quality Improvement. A number of papers concentrate on improving the accuracy of the models. Some approaches learn more models to be used in concert (ensembles) or in a randomized fashion (meta heuristics).

Ensembles use multiple models to obtain better predictive performance than could be obtained from any of the constituent models. The most notable examples are bootstrap aggregating or bagging [5], boosting [26], and random forests [6]. However, these methods do not result in a single interpretable model.

Learners based on genetic algorithms are naturally parallelizable by parallel execution of independent copies of a genetic algorithm.This results in improved accuracy [29]. These methods have a randomized approach to search space exploration, which is contradictory to the goal of Widening.

Parallel Local Search. Different local search metaheuristics include simulated annealing [21],metaheuristic approach for approximating the global optimization in a large, typically discrete, search space, tabu search in [12–15, 24]. Large neighborhood search methods [19] aim to explore complex neighborhood structures efficiently by the use of appropriate heuristics, which do not require full explicit enumeration.

Parallel variants of neighborhood-based metaheuristics [30] and the state-of-the-art surveys [7,10], as well as the book [2], which present the recent developments in the field of parallel metaheuristics. Most parallel approaches are focused on improving the running time of the sequential approaches. Other parallel approaches are focused on improving the running time and the solution quality, however, they use communication between parallel workers.

The type 1 source of parallelism is usually found within an iteration of the heuristic method, where possible moves are evaluated in parallel and the best one is selected. This strategy of low-level parallelism is only focused on improving the efficiency of the computation. It does not aim at achieving better exploration and, as a consequence, higher quality of the obtained solution.

The type 2 approach achieves parallelism by partitioning the set of decision variables. The partitioning reduces the size of the solution space, but needs to be repeated to allow the exploration of the complete solution space. This parallelization approach leads to exploring different solutions from the solutions explored in the sequential version of the same heuristic method. This approach still bears no resemblance to Widening in terms of its goals and implementations.

In type 3, the parallelism is obtained from multiple concurrent explorations of the solution space. One can differentiate several subtypes of this multiple walk approach, based on the lack or presence of cooperation, synchronization between workers and others. Type 3 approaches, also called multiple walks, or multiple runs bear similarity to Widening, and they explicitly have a goal to also increase the search space exploration, and solution quality, although in most cases this is a secondary goal.

In [7], the goal of improving the solution quality as a result of parallel multistart heuristics is explicitly stated and reported in several parallelization cases. Evaluation of the neighboring solutions in parallel leads to improvement in efficiency, while the concurrent exploration of the search space often leads to an improved solution quality. In [2], parallel metaheuristics are viewed as a separate class of heuristics all together. Often the parallel implementation does not return the same solution as the sequential implementation and the authors suggest that evaluation criteria based on the notion of solution quality have to be added to speedup measures, when evaluating parallel metaheuristics.

The biggest progress in this field has been made with the approaches that are based on cooperative multiple walks. They can be synchronous and asynchronous in nature. These approaches bear similarity with some of the motivations behind Widening. The idea about better solution quality and better search space exploration are clearly stated. In general, these implementations outperform the serial methods in solution quality. However, synchronous cooperative approaches create a lot of overhead, due to the need of constant communication. Hybrid metaheuristics can be both, multiple independent runs (MIRs), in which there is no cooperation between the parallel workers, and cooperative multistart searches. This type of approaches for multistart runs have shown the best performance with respect to efficiency and solution quality. Path relinking and scatter search [16] are two approaches commonly used in hybridization of metaheuristics, which use long term memory in order to direct the search into promising areas of the search space.

When developing strategies for Widening, we are interested in sophisticated strategies that are focused on structured search space exploration, especially ones without communication. On the other hand, in the standard MIR approaches the only thing done to prevent the parallel workers from investigating the same solutions is a different starting point. In newer strategies, multiple runs are used, with the best solution as a seed. While a lot of progress has been made in search space exploration, especially when it comes to the cooperative multiple walks, they are focused on improving the exploration via exchanging information or via adding randomization/genetic search hybridization approaches. Even more sophisticated strategies, such as path relinking, while leading to improved quality solution, are not taking advantage of the parallel compute resources for better exploration.

9. Conclusion and Future Work. From a theoretical perspective, the number of parallel workers necessary to guarantee for N_k^o a performance equivalent to that of Top-k is, generally speaking, very large. However, the experimental demonstrations show that the difference in solution quality is not that large between the two methods. This can be due to the fact that the greedy algorithm for the set cover problem is already known to perform well. Still, given a sufficient number of parallel resources and good partitioning strategy, the need for communication decreases. The experimental results show a vast difference between the runtime of the two methods, neighborhood-based Widening and Top-k Widening. The synchronized communication between the parallel workers indeed produces great overhead, as expected. The running time is not affected by the size of the neighborhood in this type of neighborhood, optimality neighborhood, because this neighborhood is built using a distance measure based on the model quality ψ. This is not true of other types of neighborhoods, where at each step not only a quality measure is evaluated, but also another distance measure, which has additional computational costs. The flaws of Widening via optimality neighborhoods 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 cause the search to focus on one area of the search space. This is also a potential flaw of Top-k. Both, Widening via optimality neighborhoods and Top-k, benefit from diversity, which helps to broaden the search and prevent the exploration of very similar solutions. Apart from explicitly using diversity, the neighborhood-based Widening approach, N_k^o, can further benefit from taking into consideration the specific topology of a given search space. This can be done either *a priori* by looking at the topology of the space of model fragments (used by the refinement operator to build new models) or by a reactive search strategy which assesses the probability distribution of intersections during the search and responds to it.

REFERENCES

[1] Akbar Z., V. N. Ivanova, M. R. Berthold. Parallel Data Mining Revisited. Better, Not Faster. In: J. Hollmén, F. Klawonn, A. Tucker (eds). Proceedings of Advances in Intelligent Data Analysis XI, IDA 2012, Helsinki, Finland, October 25–27, 2012. *Lecture Notes in Computer Science*, **7619** (2012), 33–37.

[2] Alba E. Parallel Metaheuristics: A New Class of Algorithms. Wiley, 2005.

[3] Beasley J. E. OR-Library: Distributing Test Problems by Electronic Mail. *Journal of the Operational Research Society*, **41** (1990), 1069–1072.

[4] Berthold M. R., N. Cebron, F. Dill, T. R. Gabriel, T. Kötter, T. Meinl, P. Ohl, C. Sieb, K. Thiel, B. Wiswedel. KNIME: The

Konstanz Information Miner. In: C. Preisach, H. Burkhardt, L. Schmidt-Thieme, R. Decker (eds). Data Analysis, Machine Learning and Applications, Springer Berlin Heidelberg, 2008, 319–326.

[5] Breiman L. Bagging Predictors. *Machine Learning*, **24** (1996), No 2, 123–140.

[6] Breiman L. Random Forests. *Machine Learning*, **45** (2001), No 1, 5–32.

[7] Crainic T. G., M. Toulouse. Parallel Strategies for Meta-Heuristics. In: F. Glover, G. A. Kochenberger (eds). Handbook of Metaheuristics, Springer US, Boston, MA, 2003, 475–513.

[8] Darlington J., Y.-k. Guo, J. Sutiwaraphun, H. W. To. Parallel induction algorithms for data mining. In: X. Liu, P. Cohen, M. Berthold (eds). Advances in Intelligent Data Analysis Reasoning about Data. *Lecture Notes in Computer Science*, **1280** (1997), 437–445.

[9] Dhillon I. S., D. S. Modha. A Data-Clustering Algorithm on Distributed Memory Multiprocessors. In: M. J. Zaki, C. Ho (eds). Large-Scale Parallel Data Mining, Workshop on Large-Scale Parallel KDD Systems, SIGKDD, August 15, 1999, San Diego, CA, USA, revised papers. *Lecture Notes in Computer Science*, **1759** (1999), Springer, 245–260.

[10] Gabriel T. Parallel Meta-heuristic Search. In: R. Martí, P. Panos, M. G. C. Resende (eds). Handbook of Heuristics, Springer International Publishing, Cham, 2016, 1–39.

[11] Garg A., A. Mangla, N. Gupta, V. Bhatnagar. PBIRCH: A Scalable Parallel Clustering Algorithm for Incremental Data. In: Proceedings of the 10th International Database Engineering and Applications Symposium, IEEE Computer Society, 2006, 315–316.

[12] Glover F. Future Paths for Integer Programming and Links to Artificial Intelligence. *Computers and Operations Research*, **13** (1986), No 5, 533–549.

[13] Glover F. Tabu Search—Part II. *INFORMS Journal on Computing*, **2** (1990), No 1, 4–32.

[14] Glover F. Ejection Chains, Reference Structures and Alternating Path Methods for Traveling Salesman Problems. *Discrete Applied Mathematics*, **65** (1996), No 1–3, 223–253.

[15] Glover F., M. Laguna. Tabu Search. Kluwer Academic Publishers, Norwell, MA, USA, 1997.

[16] Glover F. W., M. Laguna, R. Martí. Fundamentals of Scatter Search and Path Relinking. *Control and Cybernetics*, **29** (2000), 653–684.

[17] Ivanova V., M. R. Berthold. Diversity-Driven Widening. In: A. Tucker, F. Höppner, A. Siebes, S. Swift (eds). Proceedings of the 12th International Symposium on Intelligent Data Analysis (IDA 2013), Springer Berlin Heidelberg, 2013, 223–236.

[18] Johnson D. S. Approximation Algorithms for Combinatorial Problems. In: A. V. Aho, A. Borodin, R. L. Constable, R. W. Floyd, M. A. Harrison, R. M. Karp, H. R. Strong (eds). STOC '73: Proceedings of the Fifth Annual ACM Symposium on Theory of Computing, ACM, 1973, 38–49.

[19] Judd D., P. K. McKinley, A. K. Jain. Large-scale parallel data clustering. *TPAMI*, **20** (1998), No 8, 871–876.

[20] Kargupta H., P. Chan. Advances in Distributed and Parallel Knowledge Discovery. AAAI Press/MIT Press, 2000.

[21] Kirkpatrick S., C. D. Gelatt, M. P. Vecchi. Optimization by Simulated Annealing. *SCIENCE*, **220** (1983), No 4598, 671–680.

[22] Kufrin R. Decision Trees on Parallel Processors. In: Parallel Processing for Artificial Intelligence 3, 1997, 279–306.

[23] Kumar V., S. Ranka, V. Singh. Special Issue on High-Performance Data Mining. *Journal of Parallel and Distributed Computing*, **61** (2001), No 3, 281–284.

[24] Laguna M., J. W. Barnes, F. Glover. Intelligent Scheduling with Tabu Search: An application to jobs with linear delay penalties and sequence-dependent setup costs and times. *Appl. Intell.*, **3** (1993), No 2, 159–172.

[25] Olson C. F. Parallel Algorithms for Hierarchical Clustering. *Parallel Computing*, **21** (1995), No 8, 1313–1325.

[26] Schapire R. E. The Strength of Weak Learnability. *Machine Learning*, **5** (1990), 28–33.

[27] Shafer J. C., R. Agrawal, M. Mehta. SPRINT: A Scalable Parallel Classifier for Data Mining. In: T. M. Vijayaraman, A. P. Buchmann, C. Mohan, N. L. Sarda (eds). VLDB'96, Proceedings of 22th International Conference on Very Large Data Bases, September 3–6, 1996, Mumbai (Bombay), India, Morgan Kaufmann, 1996, 544–555.

[28] Srivastava A., E.-H. Han, V. Kumar, V. Singh. Parallel Formulations of Decision-Tree Classification Algorithms. *Data Mining and Knowledge Discovery*, **3** (1999), No 3, 237–261.

[29] Talia D. Parallelism in Knowledge Discovery Techniques. In: J. Fagerholm, J. Haataja, J. Järvinen, M. Lyly, P. Råback, V. Savolainen (eds). Applied Parallel Computing (2002), Springer Berlin Heidelberg, 127–136.

[30] Verhoeven M. G. A., E. H. Aarts. Parallel Local Search. *Journal of Heuristics*, **1** (1995), No 1, 43–65.

[31] Zaki M. J. Parallel and Distributed Association Mining: a Survey. *IEEE Concurrency*, **7** (1999), No 4, 14–25.

[32] Zaki M. J., C. Ho, R. Agrawal. Parallel Classification for Data Mining on Shared-Memory Multiprocessors. In: M. Kitsuregawa, M. P. Papazoglou, C. Pu (eds). Proceedings of the 15th International Conference on Data Engineering, Sydney, Australia, March 23–26, 1999, IEEE Computer Society, 1999, 198–205.

[33] Zaki M. J., C.-T. Ho. Large-Scale Parallel Data Mining. Springer, 2000.

[34] Zaki M. J., Y. Pan. Introduction: Recent Developments in Parallel and Distributed Data Mining. *Distributed and Parallel Databases*, **11** (2002), No 2, 123–127.

Violeta N. Ivanova-Rohling
Institute of Mathematics and Informatics
Department of Mathematical Foundations of Computer Science
Bulgarian Academy of Sciences
Acad. G. Bonchev Str., Bl. 8
1113 Sofia, Bulgaria
e-mail: `violeta@math.bas.bg`

Received June 14, 2018
Final Accepted January 12, 2022