RWTH Aachen

Department of Computer Science
Technical Report

A Faster Algorithm for the Steiner Tree Problem

Daniel Mölle, Stefan Richter, Peter Rossmanith

ISSN 0935–3232 · Aachener Informatik Berichte · AIB-2005-04

RWTH Aachen · Department of Computer Science · March 2005

The publications of the Department of Computer Science of *RWTH Aachen University* are accessible through the World Wide Web in general.

http://aib.informatik.rwth-aachen.de/

A Faster Algorithm for the Steiner Tree Problem*

Daniel Mölle, Stefan Richter, Peter Rossmanith

Lehr- und Forschungsgebiet Theoretische Informatik
RWTH Aachen, Germany
Email: {moelle,richter,rossmani}@cs.rwth-aachen.de

Abstract. The best algorithm for the Steiner tree problem by Dreyfus and Wagner is 33 years old. We celebrate this occasion and present a variation on this theme. A new algorithm is developed, which improves the running time from $O(3^k \cdot n^3)$ to $O\big((2+\varepsilon)^k \cdot poly(n)\big)$.

Dieses Baums Blatt, der von Osten
Meinem Garten anvertraut,
Gibt geheimen Sinn zu kosten,
Wie's den Wissenden erbaut.

— J.W. v. Goethe

1 Prologue

It is commonly believed in the scientific community that NP-hard problems cannot be solved in polynomial time. Nevertheless, we have to deal with many of these in everyday applications. Earlier scholarship has addressed this dilemma in a number of ways, among them approximation, randomized algorithms, parameterized complexity, heuristics and many more. Recently, there has been renewed vigor in the field of exact algorithms, and the exponential run-time bounds for many problems have been improved. Some remarkable examples for such improvement have been achieved with new algorithms for 3-Satisfiability [4], Dominating Set [6] and Max-Cut [3].

The Steiner tree problem on networks is to find a subgraph of minimum total edge weight that connects all nodes in a given node subset. Since we assume positive weights for all edges, this subgraph must be a tree. The respective decision problem, in which we ask for the existence of such a subgraph whose weight does not exceed a given limit, is known to be NP-complete. The optimization problem is also APX-complete, even if the edge weights are restricted to $\{1, 2\}$ [1]. On the positive side, it is approximable within $1 + (\ln 3)/2$, or within 1.28 in the aforementioned restricted variation [7].

The best exact algorithm for the Steiner tree problem known today is due to Dreyfus and Wagner [2]. Given a graph with n nodes, a set of k terminals and a function encoding the lengths of the edges, it computes a minimum Steiner tree in time $O(3^k n + 2^k n^2 + n^2 \log n + nm)$ using dynamic programming. In this paper, we present an algorithm achieving the same in time $O(c^k \cdot poly(n))$ for some $c < 3$.

* Supported by the DFG under grant RO 927/6-1 (TAPI)

In fact, we can prove bounds of $O(c^k \cdot poly(n))$ for arbitrary $2 < c < 3$, with large constants hidden in the Landau notation for small values of c.

For a terminal set Y of size k, the Dreyfus-Wagner algorithm works as follows. It first computes optimal Steiner trees for all subsets of Y of size two, then it uses these results to compute optimal Steiner trees for all subsets of Y of size three, and so on, until an optimal Steiner tree for Y itself is found. The crucial lemma behind this algorithm introduces the Dreyfus-Wagner recursion (as it is called in [5]), giving a strategy to find an optimal Steiner tree for a given terminal set quickly by combining two optimal Steiner trees for two subsets with specific properties.

One of the basic ideas used in the Dreyfus-Wagner recursion is the following: if T is an optimal Steiner tree for some terminal set $Y \cup \{v\}$ and v is not a leaf in T, we can split T at v to get optimal Steiner trees T_1 and T_2 for $Y' \cup \{v\}$ and $(Y - Y') \cup \{v\}$, where Y' is the set of terminals in T_1. In this paper, we will extend this concept as follows: instead of splitting the tree at a single node v, we choose an entire set X of nodes to split the tree at, where $|X|$ is bounded by $1/\varepsilon$ for an adjustable parameter ε.

An important fact is that there exists such a set X for which each of the resulting subtrees contains at most $\varepsilon|Y|+1$ of the terminals from Y. Using some additional ideas, this allows for a dynamic programming algorithm that successively combines locally optimal Steiner trees, constructing a globally optimal Steiner tree for the terminal set $Y \cup X$. In the case that some optimal Steiner tree for Y contains all the nodes from X, the constructed tree is also an optimal Steiner tree for Y.

2 Chopping trees

As already stated in the introduction, our algorithm relies on the fact that there exists a set X of nodes to split the tree at such that the resulting subtrees contain no more than $\varepsilon|Y|$ terminals each. This is formalized by the following lemma.

Lemma 1. *Let $\varepsilon > 0$, and let T be a tree with root r and $k \geq 1/\varepsilon^2$ terminals. Then there exists a set X of nodes such that*

1. *the subtree rooted at any child of any node $x \in X \cup \{r\}$ has at most εk terminals that are not part of any subtree rooted at a child of some successor $x' \in X - \{x\}$ of x, and*
2. *the size of X does not exceed $\lceil 1/\varepsilon \rceil$.*

Proof. The following algorithm computes a set X that fulfills the conditions described in the lemma:

1. Let $X := \emptyset$.
2. If T contains no more than εk terminals, output X, and stop.
3. Otherwise, let $v := r$.
4. For all the children $c_1, \ldots, c_d$ of v, count the amount of terminals contained in the subtrees rooted at $c_1, \ldots, c_d$, and let $m(v)$ be the maximum of these amounts.
5. If $m(v) \leq \varepsilon k$, add v to X, remove all the subtrees rooted at $c_1, \ldots, c_d$, and go to step 2.

6. Otherwise, in the case that $m(v) > \varepsilon k$, let v be one of the c_i with a maximal amount of terminals in the respective subtree, and go to step 4.

It is easy to see that the algorithm terminates on every input. Clearly, the output X satisfies the first condition of the lemma, as the algorithm only adds a node to X if the subtrees rooted at children of that node contain a sufficiently small amount of terminals that have not yet been processed. By cutting off processed subtrees, as done in step 5, the algorithm just avoids to count terminals that lie in subtrees of children of previously chosen nodes.

To see that the second condition is fulfilled, note that the subtree rooted at the currently inspected node v (in steps 4 through 6) always contains more than εk unprocessed terminals: otherwise, a parent of v would have been chosen as a node to cut at. Hence, when v is added to X in step 5 and all the subtrees rooted at children of v are cut off, more than $\varepsilon k - 1$ terminals disappear, where the term -1 is included because v can be one of the terminals itself. This processing of terminals continues while more than εk terminals remain. In total, and because $\varepsilon \geq \varepsilon/k$, we get that

$$|X| \leq \left\lceil \frac{k - \varepsilon k}{\varepsilon k - 1} \right\rceil = \left\lceil \frac{1 - \varepsilon}{\varepsilon - 1/k} \right\rceil = \left\lceil \frac{1}{\varepsilon} \frac{1 - \varepsilon}{1 - 1/\varepsilon k} \right\rceil \leq \left\lceil \frac{1}{\varepsilon} \right\rceil.$$

□

For an example, see Figure 1. Without loss of generality, we assume that henceforth $1/\varepsilon$ be an integer.

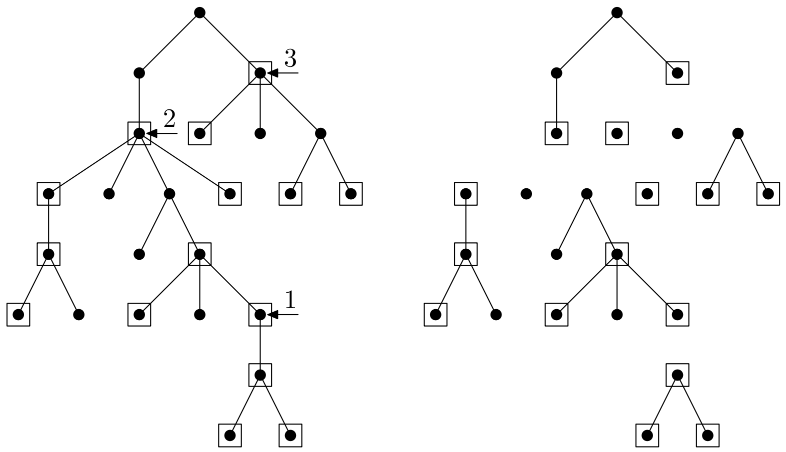

Fig. 1. Example for Lemma 1 ($k = 15$, $\varepsilon = \frac{1}{5}$, $\varepsilon k = 3$; $|X| = 3 \leq 5$)

We intend to group nodes from X in so-called *links*, whenever they serve to delimit a subtree (together with some terminals).

Definition 1. Let X be a set. A multiset $\mathcal{X} = \{\mathcal{X}_1, \ldots, \mathcal{X}_r\}$ with $\mathcal{X}_1, \ldots, \mathcal{X}_r \in 2^X \setminus \{\emptyset\}$ is called a *chain mail for* X if $|\mathcal{X}_i \cap \mathcal{X}_j| \leq 1$ for any $1 \leq i < j \leq r$ and $G_{\mathcal{X}} := (\{1, \ldots, r\}, E_{\mathcal{X}})$, $E_{\mathcal{X}} := \{\, \{i, j\} \mid \mathcal{X}_i \neq \mathcal{X}_j \wedge \mathcal{X}_i \cap \mathcal{X}_j \neq \emptyset \,\}$, is a tree. The elements of a chain mail are called *links*.

Note that a chain mail is a special kind of multiset, in that two links $\mathcal{X}_i$, $\mathcal{X}_j$ may be identical only if they are singletons. This makes sense in our context as can be seen from the way we create chain mails in Corollary 1.

To build up the final tree T , we first want to look at subtrees T_i that span a link $\mathcal{X}_i$ and some subset Y_i of the terminals. Naïvely, we might be inclined to simply select the subtrees cut off in Lemma 1; unfortunately, there might be too many of these to try them all out, since multiple links in a chain mail are allowed to contain the same set of nodes from X. Thus, in the next lemma, we create bigger subtrees by unifying several links.

Lemma 2. *Let $\varepsilon > 0$, and let T be a rooted tree with $k \geq 1/\varepsilon^2$ terminals. Moreover, let X be the set given by the algorithm used in the proof of Lemma 1. The subtrees of T, as cut off by the algorithm, can be mapped to $3/\varepsilon$ containers such that all the subtrees in each container were cut off at the same node from X and contain together at most εk terminals.*

Proof. Let us extend the algorithm presented in the proof of Lemma 1. Whenever a node is chosen to be added to X, some amount s of subtrees with at most εk terminals gets cut off the remaining tree. Create s new containers, and map each subtree onto one of them. While there are two containers with no more than $\frac{1}{2}\varepsilon k$ terminals in their subtrees, unite them. When this loop terminates, at most one of the remaining containers has less than $\frac{1}{2}\varepsilon k$ terminals in its subtrees.

In total, at most εk nodes are added to X, and hence, at most εk containers with less than $\frac{1}{2}\varepsilon k$ terminals in their subtrees remain. All other containers have more than $\frac{1}{2}\varepsilon k$ terminals, and because there are k terminals in T, there can be no more than $k/(\frac{1}{2}\varepsilon k) = 2/\varepsilon$ of them. Altogether, when our extended algorithm terminates, at most $3/\varepsilon$ containers remain. □

Definition 2. Let $\varepsilon > 0$, and let T be a tree with terminal set Y, $|Y| \geq 1/\varepsilon^2$. A chain mail $\mathcal{X}$ is said to be a *mirror of* T if there are subsets $Y_1 \cup \ldots \cup Y_r = Y$, such that there are optimal Steiner trees T_i for the terminal sets $\mathcal{X}_i \cup Y_i$ which combine to form the entire tree T. In this case we will also say that $\mathcal{X}$ *breaks the terminal set into pieces* $Y_1 \cup \ldots \cup Y_r = Y$. We call this mirror *$q$-granular* if furthermore each subset Y_i is of size at most q.

Corollary 1. *Let $\varepsilon > 0$, and let T be a tree with $k \geq 1/\varepsilon^2$ terminals. Then there is an $(\varepsilon k + 1)$-granular mirror of T for a base set X with $|X| \leq 1/\varepsilon$, and $\mathcal{X}$ contains at most $3/\varepsilon$ links.*

Proof. In the proof of Lemma 2, add to each container its respective cut node from X, and form new links as a union of old ones in the same way we formed containers from subtrees. It is easy to see that the resulting trees are optimal Steiner trees. □

3 Running in chain mail

The following two lemmata play an important rôle for the runtime analysis of our algorithm. We estimate the number of relevant chain mails for a given set X as well as the number of steps required to compute optimal Steiner trees for some small terminal sets.

Lemma 3. *Let X be a set of nodes and $m := |X|$. The number of chain mails for X with $3m$ links is bounded by $m^{2m-2}\binom{3m-1}{m-1}$.*

Proof. Every chain mail induces a tree $G_{\mathcal{X}} = (\{\mathcal{X}_1, \ldots, \mathcal{X}_r\}, E_{\mathcal{X}})$ where $E_{\mathcal{X}} = \{\, \{i,j\} \mid \mathcal{X}_i \neq \mathcal{X}_j \wedge \mathcal{X}_i \cap \mathcal{X}_j \neq \emptyset\}$. We first prove that there are no more than m inner nodes in $G_{\mathcal{X}}$. This bound is tight, as exemplified by the chain mail $\mathcal{X} = \{\{1\}, \{1,2\}, \{2,3\}, \ldots, \{m-1, m\}, \{m\}\}$ for $X = \{1, \ldots, m\}$.

Being a tree, $G_{\mathcal{X}}$ can be constructed top-down, starting from an arbitrarily picked root. By definition of $E_{\mathcal{X}}$, we also have that every node $\mathcal{X}_i$ in $G_{\mathcal{X}}$ other than the root shares exactly one element $h(i)$ from X — imagine a hook on which the new link dangles — with its parent. It is easy to see that, at any step in the inductive construction of the tree $G_{\mathcal{X}}$, for any leaf $\mathcal{X}_i$ to be transformed into an inner node, a new leaf $\mathcal{X}_j$ with $h(j) \in \mathcal{X}_i \setminus \{h(i)\}$ must be added. Observe that $h(j)$ cannot be used to turn any other leaf into an inner node later, because $h(j)$ cannot occur in any previously attached link and will be a hook in any new link which contains this element. Now the claim follows from $|X| \leq m$.

In order to achieve a non-trivial bound on the number of chain mails, we distinguish two phases in the inductive construction of $\mathcal{X}$ and $G_{\mathcal{X}}$: adding the first m nodes, some or all of which can become inner nodes in the entire process of building the tree, and attaching $2m$ leaves. Again, we use both the notion of forming a chain mail and the notion of forming the respective tree.

For the analysis of the first phase, we estimate as follows: Select a hook for each link safe the root, and then distribute the remaining nodes from X. There are $m-1$ links to select hooks from X for, yielding m^{m-1} possibilities. Moreover, at most $m-1$ nodes remain in so far that they have not been used as hooks, and they have to be assigned to one of m links, so we can do this in m^{m-1} ways again. Altogether, the first phase comprises at most m^{2m-2} alternatives to construct the respective tree.

It remains to estimate the number of ways in which $2m$ leaves can be added to $G_{\mathcal{X}}$. As leaves are singletons (each leaf link in $\mathcal{X}$ contains exactly one element from X), we only need to count the number of ways to choose $2m$ elements from the set $\{1, \ldots, m\}$ where repetitions are allowed. These are $\binom{3m-1}{2m}$ many. All in all, we count $m^{2m-2}\binom{3m-1}{m-1}$ possibilities to generate a chain mail for X. □

Lemma 4. *Let $1 > \varepsilon > 0$, $N = (V, E, \ell)$ be a network, $Y \subseteq V$ a set of terminals with $|Y| \geq 1/\varepsilon^2$, and $\mathcal{X}$ a chain mail for some X with $|X| \leq 1/\varepsilon$. Optimal Steiner trees for all sets of the form $\mathcal{X}_i \cup Y'$ where $\mathcal{X}_i \in \mathcal{X}$ and $Y' \subseteq Y$ with $|Y'| \leq \varepsilon|Y| + 1$ can be computed in time*

$$O\big(|\mathcal{X}| \cdot (\varepsilon^{\varepsilon}(1-\varepsilon)^{1-\varepsilon})^{-|Y|} \cdot 3^{\varepsilon|Y|} \cdot n^3\big).$$

Proof. There are $|\mathcal{X}|$ links, and for each link all small subsets of Y have to be considered. By an application of the Stirling formula, these are at most

$$\binom{|Y|}{\varepsilon|Y| + 1} = O\Big(\big(\varepsilon^{\varepsilon}(1-\varepsilon)^{1-\varepsilon}\big)^{-|Y|}\Big)$$

many. For each combination of a link and such a set the Dreyfus-Wagner algorithm is called, whose running time is bounded by $O(3^{1/\varepsilon + \varepsilon|Y|+1} \cdot n^3)$ in any such case. □

4 Retrieving the Holy Grail

Having established all the preliminaries, we can now conclude with the central theorem of this paper.

Let $\mathcal{X} = \{\mathcal{X}_1, \dots, \mathcal{X}_r\}$ be a chain mail for a set of nodes X. Without loss of generality, let $\mathcal{X}_1, \dots, \mathcal{X}_r$ be a sequence of links such that its every prefix forms a tree just as $\mathcal{X}$ itself. Consider the following algorithm, which constructs a table of Steiner trees:

1. For every link $\mathcal{X}_i \in \mathcal{X}$ and every $Y' \subseteq Y$ where $|Y'| \leq \varepsilon|Y| + 1$, use the Dreyfus-Wagner algorithm to generate an optimal Steiner tree for $Y' \cup \mathcal{X}_i$ and save it into the table cell $T(Y' \cup \mathcal{X}_i)$.
2. For $s := 2$ to $|\mathcal{X}|$ do:
   Let $X' := \mathcal{X}_1 \cup \dots \cup \mathcal{X}_s$. Generate a Steiner tree with terminal set $Y' \cup X'$ for each $Y' \subseteq Y$ where $|Y'| \leq s\varepsilon|Y| + s$ as follows: Minimize the expression $\ell(T(Y'' \cup \mathcal{X}_s) + \ell(T((Y' \setminus Y'') \cup (X' \setminus \mathcal{X}_s)))$ over all $Y'' \subseteq Y'$ with $|Y''| \leq \varepsilon|Y| + 1$ and $|Y' \setminus Y''| \leq (s-1)\varepsilon|Y| + s - 1$, keeping the best resulting tree. Save this tree into the table cell $T(Y' \cup X')$.

Theorem 1. *Let $\mathcal{X}$ be an $(\varepsilon|Y| + 1)$-granular mirror of an optimal Steiner tree $\mathcal{T}$ for some terminal set Y and $\varepsilon < 0.18$. Given $\mathcal{X}$, the above algorithm computes an optimal Steiner tree for Y in time*

$$O\Big(|\mathcal{X}| \cdot 2^{|Y|} \cdot \big(\varepsilon^{\varepsilon}(1-\varepsilon)^{1-\varepsilon}\big)^{-|Y|} \cdot n^3\Big).$$

Proof. We will reuse the notation from the description of the algorithm. Obviously, the table contains only Steiner trees. The mirror $\mathcal{X}$ of $\mathcal{T}$ breaks the terminal set into pieces $Y_1 \cup \dots \cup Y_{|\mathcal{X}|} = Y$. The correctness of the algorithm follows from the fact that for each $1 \leq s \leq |\mathcal{X}|$ the trees $T(Y' \cup X')$ are optimal for $Y' = Y_1 \cup \dots \cup Y_s$ and $X' = \mathcal{X}_1 \cup \dots \cup \mathcal{X}_s$. In $\mathcal{T}$ we have an optimal Steiner tree that contains X, and thus, any optimal Steiner tree for $Y \cup X$ is a solution to the Steiner tree problem for Y.

To see the above fact, note first that it is true for $s = 1$ if we believe in the correctness of the Dreyfus-Wagner algorithm, and that furthermore all the entries computed in the first phase are optimal. For the induction step, consider $T(Y' \cup X')$ where $Y' = Y_1 \cup \dots \cup Y_s$ and $X' = \mathcal{X}_1 \cup \dots \cup \mathcal{X}_s$. As $\mathcal{X}$ is a mirror of the optimal Steiner tree $\mathcal{T}$, there is a subtree T' of $\mathcal{T}$ that is optimal for $Y' \cup X'$. In contradiction to our claim, suppose that T' is cheaper than $T(Y' \cup X')$. Obviously, $\{X_1, \dots, \mathcal{X}_s\}$ is also a mirror of T'. Therefore, we can split T' at the articulation node that is the only element in $\mathcal{X}_{s-1} \cap \mathcal{X}_s$ to obtain two optimal Steiner trees for the respective subsets of the terminals. Because $\ell(T') < \ell(T(Y' \cup X'))$, this implies that at least one of both is better than the respective table entry. This contradicts the induction hypothesis.

On the running time: A numerical analysis of the bound in Lemma 4 shows that the first phase takes less than $O(2^{|Y|} \cdot n^3)$ for $\varepsilon \leq 0.18$. Observe that the precondition $|Y| \geq 1/\varepsilon^2$ can be assumed to be fulfilled without loss of generality, since the running time is $O(poly(n))$ if $|Y|$ is bounded by a constant.

The second phase consists of $|\mathcal{X}| - 1 < |\mathcal{X}|$ steps. In each of these, all the $Y' \subseteq Y$ of limited size must be looked at. These are less than $2^{|Y|}$ sets. Again, for each Y', we need to go through all bipartitions where the size of the smaller class Y'' is at most $\varepsilon|Y|+1$. As in the proof of Lemma 4, there are no more than

$$\binom{|Y|}{\varepsilon|Y|+1} = O\Big(\big(\varepsilon^{\varepsilon}(1-\varepsilon)^{1-\varepsilon}\big)^{-|Y|}\Big)$$

of those.

Altogether, the running time of the algorithm is dominated by the second phase and thus bounded by the product of these three values, multiplied by some polynomial which does not exceed n^3. ☐

Note that this runtime bound still holds if $\mathcal{X}$ is just a chain mail, but not an $(\varepsilon|Y|+1)$-granular mirror of any optimal Steiner tree. The above algorithm may just not find an optimal Steiner tree in this case.

Corollary 2. *Let $N = (V, E, \ell)$ be a network with k terminals and $0 < \delta$. We can solve the respective Steiner tree problem in time $O\big((2+\delta)^k \cdot poly(n)\big)$.*

Proof. The crucial factor in the bound from Theorem 1 is $(\varepsilon^{\varepsilon}(1-\varepsilon)^{1-\varepsilon})^{-|Y|}$. As ε approaches 0, the base in this expression converges towards 1. Hence, there is an appropriate $\varepsilon > 0$ such that $(\varepsilon^{\varepsilon}(1-\varepsilon)^{1-\varepsilon})^{-|Y|} = O((1+\delta/2)^{|Y|})$. Using the theorem, we thus get a running time of

$$O\big(|\mathcal{X}| \cdot (2+\delta)^{|Y|} \cdot n^3\big)$$

for each chain mail we consider.

By Corollary 1, there exists a $(1/\varepsilon+1)$-granular mirror $\mathcal{X}$ of size at most $3/\varepsilon$ for an optimal Steiner tree in our network, whose base set X satisfies $|X| \leq 1/\varepsilon$. Unfortunately, we do not know how to choose $\mathcal{X}$ or even X. However, by Lemma 3, there are no more than

$$O\Big(|X|^{2|X|-2}\binom{3|X|-1}{|X|-1}\Big)$$

different chain mails for a fixed X. Clearly, there are less than $n^{1/\varepsilon}$ different sets $X \subseteq V$ with $|X| \leq 1/\varepsilon$. Thus the total number of chain mails to go through is $O(n^{1/\varepsilon})$. In total, we get an asymptotic running time of

$$O\big(n^{1/\varepsilon} \cdot |\mathcal{X}| \cdot (2+\delta)^{|Y|} \cdot n^3\big) = O\big((2+\delta)^k \cdot poly(n)\big).$$

☐

5 Valor and sincerity

In this section, we take a closer look at the influence of ε on the running time.

Corollary 2 states that the Steiner tree problem can be solved in time $O(c^k \cdot poly(n))$ for any $c > 2$ (given that k is sufficiently large, as $\varepsilon k \geq 1$ is a precondition for Lemma 1). However, this bound obscures the combinatorial explosion occuring as ε approaches 0, and we emphatically point out that we cannot expect the algorithm to be applicable for very small ε since it may have to go through too many chain mails.

On the other hand, the estimation on the number of chain mails in Lemma 3 is rather rough. In particular, the analysis of the first phase does not exploit the fact that only those nodes from the base set that were not used as hooks need to be distributed to the links later. Moreover, we consider chain mails $\mathcal{X}$ of up to $3/\varepsilon$ links, but we have yet to find an instance where such a huge chain mail is required as an appropriate mirror.

Let us recall some of the estimations from above, and put them together. For the sake of readability, we write a/ε instead of $\lfloor a/\varepsilon \rfloor$. By Lemma 3, the total number of chain mails to consider is bounded by

$$\binom{n}{1/\varepsilon} \cdot 1/\varepsilon^{2/\varepsilon-2} \binom{3/\varepsilon-1}{1/\varepsilon-1}.$$

We already know that the running time of the core algorithm is dominated by its second phase, as seen in the proof of Theorem 1. It performs at most

$$3/\varepsilon \cdot 2^k \cdot \left(\frac{1}{\varepsilon^\varepsilon(1-\varepsilon)^{1-\varepsilon}}\right)^k$$

many steps per chain mail.

To exemplify the combinatorial explosion, we give the resulting bounds for several values of ε. Observe that the runtime bound for the second phase given by Theorem 1 includes the expression $2^k \cdot (\varepsilon^\varepsilon(1-\varepsilon)^{1-\varepsilon})^{-k}$. This is less than 3^k for ε smaller than about 0.1402. In the following table, the number of chain mails per X determines the constant factor in the running time, while the number of node sets X determines the polynomial in the asymptotic bound.

ε	chain mails per X	total asymptotic runtime
1/8	$1.0783 \cdot 10^{18}$	$O(2.9152^k \cdot n^{11})$
1/20	$3.8409 \cdot 10^{64}$	$O(2.4392^k \cdot n^{23})$
1/50	$2.1172 \cdot 10^{208}$	$O(2.2061^k \cdot n^{53})$

The picture we come across here is not an unknown one. The tireless wanderer has encountered similar phenomena in the wide field of polynomial time approximation schemes (PTAS). In both cases, small ε lead to humongous contants and large exponents for the polynomials. The difference lies in what we try to achieve by choosing smaller and smaller ε. Whereas we try to get the base of the exponential part of the runtime bound as close to two as possible, approxionados crave optimality.

6 Epilogue

In terms of parameterized complexity, the algorithm presented in this paper is the asymptotically fastest solution to the Steiner tree problem today. While our algorithm is only a bit more complicated than the one by Dreyfus and Wagner, the constants hidden in the Landau notation can become very large if ε is chosen too small. This is because we have to go through all the chain mails with $\lfloor 3/\varepsilon \rfloor$ links for all the $\lfloor 1/\varepsilon \rfloor$-element subsets of V. While there are only $\binom{n}{\lfloor 1/\varepsilon \rfloor}$ ways to choose an $X \subseteq V$ with $|X| = \lfloor 1/\varepsilon \rfloor$, the number of relevant chain mails grows very rapidly in $1/\varepsilon$, as detailed above.

Hence, in practice, our algorithm is very likely to be slower than the Dreyfus-Wagner algorithm for small values of n or k (or if ε is chosen badly). On the other hand, there is plenty potential for improvement, such as omitting sets X or chain mails $\mathcal{X}$ that cannot lead to optimal solutions. To enable more precise statements, work is underway on implementing both algorithms for a comparative analysis. Furthermore, we plan on reducing the number of nodes in X and of chain mails $\mathcal{X}$ by dynamically adapting the amount of terminals a link is responsible for.

One might imagine that the basic technique employed lends itself to generalization. That is, it should be worthwhile to look at graph algorithms using a dynamic programming approach similar to the Dreyfus-Wagner one, and try to choose more than one point of combination in order to balance the size of subproblems.

Acknowledgement. The third author wishes to thank Rolf Niedermeier for suggesting this problem.

Knights of the LuFGTI

Daniel “Gurnemanz” Mölle
Stefan “Gawan” Richter
Peter “Parzival” Rossmanith

Historical Note. Did you know? When we weigh the authors' years of birth according to their respective ratio of contribution, their average yields exactly the date the seminal paper by Dreyfus and Wagner was published.

References

1. M. Bern and P. Plassmann. The steiner problem with edge lengths 1 and 2. *Information Processing Letters*, 32:171–176, 1989.
2. S. E. Dreyfus and R. A. Wagner. The steiner problem in graphs. *Networks*, 1:195–207, 1972.
3. S. S. Fedin and A. S. Kulikov. A $2^{|E|/4}$-time algorithm for max-cut. In *Zapiski nauchnyh seminarov POMI*, volume 293, pages 129–138, 2002. English translation is to appear in Journal of Mathematical Sciences.
4. K. Iwama. and S. Tamaki. Improved upper bounds for 3-sat. Technical Report TR03-053, ECCC Trier, 2003.
5. H. J. Prömel and A. Steger. *The Steiner Tree Problem.* Vieweg, 2002.
6. B. Randerath and I. Schiermeyer. Exact algorithms for minimum dominating set. Technical report, Zentrum für Angewandte Informatik Köln, Lehrstuhl Speckenmeyer, April 2004.
7. G. Robins and A. Zelikovsky. Improved steiner tree approximation in graphs. In *Proceedings of the 11th ACM-SIAM Symposium on Discrete Algorithms*, pages 770–779, 2000.

Aachener Informatik-Berichte

This is a list of recent technical reports. To obtain copies of technical reports please consult http://aib.informatik.rwth-aachen.de/ or send your request to: Informatik-Bibliothek, RWTH Aachen, Ahornstr. 55, 52056 Aachen, Email: `biblio@informatik.rwth-aachen.de`

1987-01 *	Fachgruppe Informatik: Jahresbericht 1986
1987-02 *	David de Frutos Escrig, Klaus Indermark: Equivalence Relations of Non-Deterministic Ianov-Schemes
1987-03 *	Manfred Nagl: A Software Development Environment based on Graph Technology
1987-04 *	Claus Lewerentz, Manfred Nagl, Bernhard Westfechtel: On Integration Mechanisms within a Graph-Based Software Development Environment
1987-05 *	Reinhard Rinn: Über Eingabeanomalien bei verschiedenen Inferenzmodellen
1987-06 *	Werner Damm, Gert Döhmen: Specifying Distributed Computer Architectures in AADL*
1987-07 *	Gregor Engels, Claus Lewerentz, Wilhelm Schäfer: Graph Grammar Engineering: A Software Specification Method
1987-08 *	Manfred Nagl: Set Theoretic Approaches to Graph Grammars
1987-09 *	Claus Lewerentz, Andreas Schürr: Experiences with a Database System for Software Documents
1987-10 *	Herbert Klaeren, Klaus Indermark: A New Implementation Technique for Recursive Function Definitions
1987-11 *	Rita Loogen: Design of a Parallel Programmable Graph Reduction Machine with Distributed Memory
1987-12	J. Börstler, U. Möncke, R. Wilhelm: Table compression for tree automata
1988-01 *	Gabriele Esser, Johannes Rückert, Frank Wagner: Gesellschaftliche Aspekte der Informatik
1988-02 *	Peter Martini, Otto Spaniol: Token-Passing in High-Speed Backbone Networks for Campus-Wide Environments
1988-03 *	Thomas Welzel: Simulation of a Multiple Token Ring Backbone
1988-04 *	Peter Martini: Performance Comparison for HSLAN Media Access Protocols
1988-05 *	Peter Martini: Performance Analysis of Multiple Token Rings
1988-06 *	Andreas Mann, Johannes Rückert, Otto Spaniol: Datenfunknetze
1988-07 *	Andreas Mann, Johannes Rückert: Packet Radio Networks for Data Exchange
1988-08 *	Andreas Mann, Johannes Rückert: Concurrent Slot Assignment Protocol for Packet Radio Networks
1988-09 *	W. Kremer, F. Reichert, J. Rückert, A. Mann: Entwurf einer Netzwerktopologie für ein Mobilfunknetz zur Unterstützung des öffentlichen Straßenverkehrs
1988-10 *	Kai Jakobs: Towards User-Friendly Networking
1988-11 *	Kai Jakobs: The Directory - Evolution of a Standard
1988-12 *	Kai Jakobs: Directory Services in Distributed Systems - A Survey
1988-13 *	Martine Schümmer: RS-511, a Protocol for the Plant Floor

1988-14 * U. Quernheim: Satellite Communication Protocols - A Performance Comparison Considering On-Board Processing

1988-15 * Peter Martini, Otto Spaniol, Thomas Welzel: File Transfer in High Speed Token Ring Networks: Performance Evaluation by Approximate Analysis and Simulation

1988-16 * Fachgruppe Informatik: Jahresbericht 1987

1988-17 * Wolfgang Thomas: Automata on Infinite Objects

1988-18 * Michael Sonnenschein: On Petri Nets and Data Flow Graphs

1988-19 * Heiko Vogler: Functional Distribution of the Contextual Analysis in Block-Structured Programming Languages: A Case Study of Tree Transducers

1988-20 * Thomas Welzel: Einsatz des Simulationswerkzeuges QNAP2 zur Leistungsbewertung von Kommunikationsprotokollen

1988-21 * Th. Janning, C. Lewerentz: Integrated Project Team Management in a Software Development Environment

1988-22 * Joost Engelfriet, Heiko Vogler: Modular Tree Transducers

1988-23 * Wolfgang Thomas: Automata and Quantifier Hierarchies

1988-24 * Uschi Heuter: Generalized Definite Tree Languages

1989-01 * Fachgruppe Informatik: Jahresbericht 1988

1989-02 * G. Esser, J. Rückert, F. Wagner (Hrsg.): Gesellschaftliche Aspekte der Informatik

1989-03 * Heiko Vogler: Bottom-Up Computation of Primitive Recursive Tree Functions

1989-04 * Andy Schürr: Introduction to PROGRESS, an Attribute Graph Grammar Based Specification Language

1989-05 J. Börstler: Reuse and Software Development - Problems, Solutions, and Bibliography (in German)

1989-06 * Kai Jakobs: OSI - An Appropriate Basis for Group Communication?

1989-07 * Kai Jakobs: ISO's Directory Proposal - Evolution, Current Status and Future Problems

1989-08 * Bernhard Westfechtel: Extension of a Graph Storage for Software Documents with Primitives for Undo/Redo and Revision Control

1989-09 * Peter Martini: High Speed Local Area Networks - A Tutorial

1989-10 * P. Davids, Th. Welzel: Performance Analysis of DQDB Based on Simulation

1989-11 * Manfred Nagl (Ed.): Abstracts of Talks presented at the WG '89 15th International Workshop on Graphtheoretic Concepts in Computer Science

1989-12 * Peter Martini: The DQDB Protocol - Is it Playing the Game?

1989-13 * Martine Schümmer: CNC/DNC Communication with MAP

1989-14 * Martine Schümmer: Local Area Networks for Manufactoring Environments with hard Real-Time Requirements

1989-15 * M. Schümmer, Th. Welzel, P. Martini: Integration of Field Bus and MAP Networks - Hierarchical Communication Systems in Production Environments

1989-16 * G. Vossen, K.-U. Witt: SUXESS: Towards a Sound Unification of Extensions of the Relational Data Model

1989-17 * J. Derissen, P. Hruschka, M.v.d. Beeck, Th. Janning, M. Nagl: Integrating Structured Analysis and Information Modelling
1989-18 A. Maassen: Programming with Higher Order Functions
1989-19 * Mario Rodriguez-Artalejo, Heiko Vogler: A Narrowing Machine for Syntax Directed BABEL
1989-20 H. Kuchen, R. Loogen, J.J. Moreno Navarro, M. Rodriguez Artalejo: Graph-based Implementation of a Functional Logic Language
1990-01 * Fachgruppe Informatik: Jahresbericht 1989
1990-02 * Vera Jansen, Andreas Potthoff, Wolfgang Thomas, Udo Wermuth: A Short Guide to the AMORE System (Computing Automata, MOnoids and Regular Expressions)
1990-03 * Jerzy Skurczynski: On Three Hierarchies of Weak SkS Formulas
1990-04 R. Loogen: Stack-based Implementation of Narrowing
1990-05 H. Kuchen, A. Wagener: Comparison of Dynamic Load Balancing Strategies
1990-06 * Kai Jakobs, Frank Reichert: Directory Services for Mobile Communication
1990-07 * Kai Jakobs: What's Beyond the Interface - OSI Networks to Support Cooperative Work
1990-08 * Kai Jakobs: Directory Names and Schema - An Evaluation
1990-09 * Ulrich Quernheim, Dieter Kreuer: Das CCITT - Signalisierungssystem Nr. 7 auf Satellitenstrecken; Simulation der Zeichengabestrecke
1990-11 H. Kuchen, R. Loogen, J.J. Moreno Navarro, M. Rodriguez Artalejo: Lazy Narrowing in a Graph Machine
1990-12 * Kai Jakobs, Josef Kaltwasser, Frank Reichert, Otto Spaniol: Der Computer fährt mit
1990-13 * Rudolf Mathar, Andreas Mann: Analyzing a Distributed Slot Assignment Protocol by Markov Chains
1990-14 A. Maassen: Compilerentwicklung in Miranda - ein Praktikum in funktionaler Programmierung (written in german)
1990-15 * Manfred Nagl, Andreas Schürr: A Specification Environment for Graph Grammars
1990-16 A. Schürr: PROGRESS: A VHL-Language Based on Graph Grammars
1990-17 * Marita Möller: Ein Ebenenmodell wissensbasierter Konsultationen - Unterstützung für Wissensakquisition und Erklärungsfähigkeit
1990-18 * Eric Kowalewski: Entwurf und Interpretation einer Sprache zur Beschreibung von Konsultationsphasen in Expertensystemen
1990-20 Y. Ortega Mallen, D. de Frutos Escrig: A Complete Proof System for Timed Observations
1990-21 * Manfred Nagl: Modelling of Software Architectures: Importance, Notions, Experiences
1990-22 H. Fassbender, H. Vogler: A Call-by-need Implementation of Syntax Directed Functional Programming
1991-01 Guenther Geiler (ed.), Fachgruppe Informatik: Jahresbericht 1990
1991-03 B. Steffen, A. Ingolfsdottir: Characteristic Formulae for Processes with Divergence
1991-04 M. Portz: A new class of cryptosystems based on interconnection networks

1991-05 H. Kuchen, G. Geiler: Distributed Applicative Arrays
1991-06 * Ludwig Staiger: Kolmogorov Complexity and Hausdorff Dimension
1991-07 * Ludwig Staiger: Syntactic Congruences for w-languages
1991-09 * Eila Kuikka: A Proposal for a Syntax-Directed Text Processing System
1991-10 K. Gladitz, H. Fassbender, H. Vogler: Compiler-based Implementation of Syntax-Directed Functional Programming
1991-11 R. Loogen, St. Winkler: Dynamic Detection of Determinism in Functional Logic Languages
1991-12 * K. Indermark, M. Rodriguez Artalejo (Eds.): Granada Workshop on the Integration of Functional and Logic Programming
1991-13 * Rolf Hager, Wolfgang Kremer: The Adaptive Priority Scheduler: A More Fair Priority Service Discipline
1991-14 * Andreas Fasbender, Wolfgang Kremer: A New Approximation Algorithm for Tandem Networks with Priority Nodes
1991-15 J. Börstler, A. Zündorf: Revisiting extensions to Modula-2 to support reusability
1991-16 J. Börstler, Th. Janning: Bridging the gap between Requirements Analysis and Design
1991-17 A. Zündorf, A. Schürr: Nondeterministic Control Structures for Graph Rewriting Systems
1991-18 * Matthias Jarke, John Mylopoulos, Joachim W. Schmidt, Yannis Vassiliou: DAIDA: An Environment for Evolving Information Systems
1991-19 M. Jeusfeld, M. Jarke: From Relational to Object-Oriented Integrity Simplification
1991-20 G. Hogen, A. Kindler, R. Loogen: Automatic Parallelization of Lazy Functional Programs
1991-21 * Prof. Dr. rer. nat. Otto Spaniol: ODP (Open Distributed Processing): Yet another Viewpoint
1991-22 H. Kuchen, F. Lücking, H. Stoltze: The Topology Description Language TDL
1991-23 S. Graf, B. Steffen: Compositional Minimization of Finite State Systems
1991-24 R. Cleaveland, J. Parrow, B. Steffen: The Concurrency Workbench: A Semantics Based Tool for the Verification of Concurrent Systems
1991-25 * Rudolf Mathar, Jürgen Mattfeldt: Optimal Transmission Ranges for Mobile Communication in Linear Multihop Packet Radio Networks
1991-26 M. Jeusfeld, M. Staudt: Query Optimization in Deductive Object Bases
1991-27 J. Knoop, B. Steffen: The Interprocedural Coincidence Theorem
1991-28 J. Knoop, B. Steffen: Unifying Strength Reduction and Semantic Code Motion
1991-30 T. Margaria: First-Order theories for the verification of complex FSMs
1991-31 B. Steffen: Generating Data Flow Analysis Algorithms from Modal Specifications
1992-01 Stefan Eherer (ed.), Fachgruppe Informatik: Jahresbericht 1991
1992-02 * Bernhard Westfechtel: Basismechanismen zur Datenverwaltung in strukturbezogenen Hypertextsystemen
1992-04 S. A. Smolka, B. Steffen: Priority as Extremal Probability
1992-05 * Matthias Jarke, Carlos Maltzahn, Thomas Rose: Sharing Processes: Team Coordination in Design Repositories

1992-06 O. Burkart, B. Steffen: Model Checking for Context-Free Processes

1992-07 * Matthias Jarke, Klaus Pohl: Information Systems Quality and Quality Information Systems

1992-08 * Rudolf Mathar, Jürgen Mattfeldt: Analyzing Routing Strategy NFP in Multihop Packet Radio Networks on a Line

1992-09 * Alfons Kemper, Guido Moerkotte: Grundlagen objektorientierter Datenbanksysteme

1992-10 Matthias Jarke, Manfred Jeusfeld, Andreas Miethsam, Michael Gocek: Towards a logic-based reconstruction of software configuration management

1992-11 Werner Hans: A Complete Indexing Scheme for WAM-based Abstract Machines

1992-12 W. Hans, R. Loogen, St. Winkler: On the Interaction of Lazy Evaluation and Backtracking

1992-13 * Matthias Jarke, Thomas Rose: Specification Management with CAD

1992-14 Th. Noll, H. Vogler: Top-down Parsing with Simultaneous Evaluation on Noncircular Attribute Grammars

1992-15 A. Schuerr, B. Westfechtel: Graphgrammatiken und Graphersetzungssysteme(written in german)

1992-16 * Graduiertenkolleg Informatik und Technik (Hrsg.): Forschungsprojekte des Graduiertenkollegs Informatik und Technik

1992-17 M. Jarke (ed.): ConceptBase V3.1 User Manual

1992-18 * Clarence A. Ellis, Matthias Jarke (Eds.): Distributed Cooperation in Integrated Information Systems - Proceedings of the Third International Workshop on Intelligent and Cooperative Information Systems

1992-19-00 H. Kuchen, R. Loogen (eds.): Proceedings of the 4th Int. Workshop on the Parallel Implementation of Functional Languages

1992-19-01 G. Hogen, R. Loogen: PASTEL - A Parallel Stack-Based Implementation of Eager Functional Programs with Lazy Data Structures (Extended Abstract)

1992-19-02 H. Kuchen, K. Gladitz: Implementing Bags on a Shared Memory MIMD-Machine

1992-19-03 C. Rathsack, S.B. Scholz: LISA - A Lazy Interpreter for a Full-Fledged Lambda-Calculus

1992-19-04 T.A. Bratvold: Determining Useful Parallelism in Higher Order Functions

1992-19-05 S. Kahrs: Polymorphic Type Checking by Interpretation of Code

1992-19-06 M. Chakravarty, M. Köhler: Equational Constraints, Residuation, and the Parallel JUMP-Machine

1992-19-07 J. Seward: Polymorphic Strictness Analysis using Frontiers (Draft Version)

1992-19-08 D. Gärtner, A. Kimms, W. Kluge: pi-Redˆ+ - A Compiling Graph-Reduction System for a Full Fledged Lambda-Calculus

1992-19-09 D. Howe, G. Burn: Experiments with strict STG code

1992-19-10 J. Glauert: Parallel Implementation of Functional Languages Using Small Processes

1992-19-11 M. Joy, T. Axford: A Parallel Graph Reduction Machine

1992-19-12 A. Bennett, P. Kelly: Simulation of Multicache Parallel Reduction

1992-19-13 K. Langendoen, D.J. Agterkamp: Cache Behaviour of Lazy Functional Programs (Working Paper)

1992-19-14 K. Hammond, S. Peyton Jones: Profiling scheduling strategies on the GRIP parallel reducer

1992-19-15 S. Mintchev: Using Strictness Information in the STG-machine

1992-19-16 D. Rushall: An Attribute Grammar Evaluator in Haskell

1992-19-17 J. Wild, H. Glaser, P. Hartel: Statistics on storage management in a lazy functional language implementation

1992-19-18 W.S. Martins: Parallel Implementations of Functional Languages

1992-19-19 D. Lester: Distributed Garbage Collection of Cyclic Structures (Draft version)

1992-19-20 J.C. Glas, R.F.H. Hofman, W.G. Vree: Parallelization of Branch-and-Bound Algorithms in a Functional Programming Environment

1992-19-21 S. Hwang, D. Rushall: The nu-STG machine: a parallelized Spineless Tagless Graph Reduction Machine in a distributed memory architecture (Draft version)

1992-19-22 G. Burn, D. Le Metayer: Cps-Translation and the Correctness of Optimising Compilers

1992-19-23 S.L. Peyton Jones, P. Wadler: Imperative functional programming (Brief summary)

1992-19-24 W. Damm, F. Liu, Th. Peikenkamp: Evaluation and Parallelization of Functions in Functional + Logic Languages (abstract)

1992-19-25 M. Kesseler: Communication Issues Regarding Parallel Functional Graph Rewriting

1992-19-26 Th. Peikenkamp: Charakterizing and representing neededness in functional loginc languages (abstract)

1992-19-27 H. Doerr: Monitoring with Graph-Grammars as formal operational Models

1992-19-28 J. van Groningen: Some implementation aspects of Concurrent Clean on distributed memory architectures

1992-19-29 G. Ostheimer: Load Bounding for Implicit Parallelism (abstract)

1992-20 H. Kuchen, F.J. Lopez Fraguas, J.J. Moreno Navarro, M. Rodriguez Artalejo: Implementing Disequality in a Lazy Functional Logic Language

1992-21 H. Kuchen, F.J. Lopez Fraguas: Result Directed Computing in a Functional Logic Language

1992-22 H. Kuchen, J.J. Moreno Navarro, M.V. Hermenegildo: Independent AND-Parallel Narrowing

1992-23 T. Margaria, B. Steffen: Distinguishing Formulas for Free

1992-24 K. Pohl: The Three Dimensions of Requirements Engineering

1992-25 * R. Stainov: A Dynamic Configuration Facility for Multimedia Communications

1992-26 * Michael von der Beeck: Integration of Structured Analysis and Timed Statecharts for Real-Time and Concurrency Specification

1992-27 W. Hans, St. Winkler: Aliasing and Groundness Analysis of Logic Programs through Abstract Interpretation and its Safety

1992-28 * Gerhard Steinke, Matthias Jarke: Support for Security Modeling in Information Systems Design

1992-29 B. Schinzel: Warum Frauenforschung in Naturwissenschaft und Technik

1992-30 A. Kemper, G. Moerkotte, K. Peithner: Object-Orientation Axiomatised by Dynamic Logic

1992-32 * Bernd Heinrichs, Kai Jakobs: Timer Handling in High-Performance Transport Systems

1992-33 * B. Heinrichs, K. Jakobs, K. Lenßen, W. Reinhardt, A. Spinner: Euro-Bridge: Communication Services for Multimedia Applications

1992-34 C. Gerlhof, A. Kemper, Ch. Kilger, G. Moerkotte: Partition-Based Clustering in Object Bases: From Theory to Practice

1992-35 J. Börstler: Feature-Oriented Classification and Reuse in IPSEN

1992-36 M. Jarke, J. Bubenko, C. Rolland, A. Sutcliffe, Y. Vassiliou: Theories Underlying Requirements Engineering: An Overview of NATURE at Genesis

1992-37 * K. Pohl, M. Jarke: Quality Information Systems: Repository Support for Evolving Process Models

1992-38 A. Zuendorf: Implementation of the imperative / rule based language PROGRES

1992-39 P. Koch: Intelligentes Backtracking bei der Auswertung funktional-logischer Programme

1992-40 * Rudolf Mathar, Jürgen Mattfeldt: Channel Assignment in Cellular Radio Networks

1992-41 * Gerhard Friedrich, Wolfgang Neidl: Constructive Utility in Model-Based Diagnosis Repair Systems

1992-42 * P. S. Chen, R. Hennicker, M. Jarke: On the Retrieval of Reusable Software Components

1992-43 W. Hans, St.Winkler: Abstract Interpretation of Functional Logic Languages

1992-44 N. Kiesel, A. Schuerr, B. Westfechtel: Design and Evaluation of GRAS, a Graph-Oriented Database System for Engineering Applications

1993-01 * Fachgruppe Informatik: Jahresbericht 1992

1993-02 * Patrick Shicheng Chen: On Inference Rules of Logic-Based Information Retrieval Systems

1993-03 G. Hogen, R. Loogen: A New Stack Technique for the Management of Runtime Structures in Distributed Environments

1993-05 A. Zündorf: A Heuristic for the Subgraph Isomorphism Problem in Executing PROGRES

1993-06 A. Kemper, D. Kossmann: Adaptable Pointer Swizzling Strategies in Object Bases: Design, Realization, and Quantitative Analysis

1993-07 * Graduiertenkolleg Informatik und Technik (Hrsg.): Graduiertenkolleg Informatik und Technik

1993-08 * Matthias Berger: k-Coloring Vertices using a Neural Network with Convergence to Valid Solutions

1993-09 M. Buchheit, M. Jeusfeld, W. Nutt, M. Staudt: Subsumption between Queries to Object-Oriented Databases

1993-10 O. Burkart, B. Steffen: Pushdown Processes: Parallel Composition and Model Checking

1993-11 * R. Große-Wienker, O. Hermanns, D. Menzenbach, A. Pollacks, S. Repetzki, J. Schwartz, K. Sonnenschein, B. Westfechtel: Das SUKITS-Projekt: A-posteriori-Integration heterogener CIM-Anwendungssysteme

1993-12 * Rudolf Mathar, Jürgen Mattfeldt: On the Distribution of Cumulated Interference Power in Rayleigh Fading Channels
1993-13 O. Maler, L. Staiger: On Syntactic Congruences for omega-languages
1993-14 M. Jarke, St. Eherer, R. Gallersdoerfer, M. Jeusfeld, M. Staudt: ConceptBase - A Deductive Object Base Manager
1993-15 M. Staudt, H.W. Nissen, M.A. Jeusfeld: Query by Class, Rule and Concept
1993-16 * M. Jarke, K. Pohl, St. Jacobs et al.: Requirements Engineering: An Integrated View of Representation Process and Domain
1993-17 * M. Jarke, K. Pohl: Establishing Vision in Context: Towards a Model of Requirements Processes
1993-18 W. Hans, H. Kuchen, St. Winkler: Full Indexing for Lazy Narrowing
1993-19 W. Hans, J.J. Ruz, F. Saenz, St. Winkler: A VHDL Specification of a Shared Memory Parallel Machine for Babel
1993-20 * K. Finke, M. Jarke, P. Szczurko, R. Soltysiak: Quality Management for Expert Systems in Process Control
1993-21 M. Jarke, M.A. Jeusfeld, P. Szczurko: Three Aspects of Intelligent Cooperation in the Quality Cycle
1994-01 Margit Generet, Sven Martin (eds.), Fachgruppe Informatik: Jahresbericht 1993
1994-02 M. Lefering: Development of Incremental Integration Tools Using Formal Specifications
1994-03 * P. Constantopoulos, M. Jarke, J. Mylopoulos, Y. Vassiliou: The Software Information Base: A Server for Reuse
1994-04 * Rolf Hager, Rudolf Mathar, Jürgen Mattfeldt: Intelligent Cruise Control and Reliable Communication of Mobile Stations
1994-05 * Rolf Hager, Peter Hermesmann, Michael Portz: Feasibility of Authentication Procedures within Advanced Transport Telematics
1994-06 * Claudia Popien, Bernd Meyer, Axel Kuepper: A Formal Approach to Service Import in ODP Trader Federations
1994-07 P. Peters, P. Szczurko: Integrating Models of Quality Management Methods by an Object-Oriented Repository
1994-08 * Manfred Nagl, Bernhard Westfechtel: A Universal Component for the Administration in Distributed and Integrated Development Environments
1994-09 * Patrick Horster, Holger Petersen: Signatur- und Authentifikationsverfahren auf der Basis des diskreten Logarithmusproblems
1994-11 A. Schürr: PROGRES, A Visual Language and Environment for PROgramming with Graph REwrite Systems
1994-12 A. Schürr: Specification of Graph Translators with Triple Graph Grammars
1994-13 A. Schürr: Logic Based Programmed Structure Rewriting Systems
1994-14 L. Staiger: Codes, Simplifying Words, and Open Set Condition
1994-15 * Bernhard Westfechtel: A Graph-Based System for Managing Configurations of Engineering Design Documents
1994-16 P. Klein: Designing Software with Modula-3
1994-17 I. Litovsky, L. Staiger: Finite acceptance of infinite words

1994-18 G. Hogen, R. Loogen: Parallel Functional Implementations: Graphbased vs. Stackbased Reduction

1994-19 M. Jeusfeld, U. Johnen: An Executable Meta Model for Re-Engineering of Database Schemas

1994-20 * R. Gallersdörfer, M. Jarke, K. Klabunde: Intelligent Networks as a Data Intensive Application (INDIA)

1994-21 M. Mohnen: Proving the Correctness of the Static Link Technique Using Evolving Algebras

1994-22 H. Fernau, L. Staiger: Valuations and Unambiguity of Languages, with Applications to Fractal Geometry

1994-24 * M. Jarke, K. Pohl, R. Dömges, St. Jacobs, H. W. Nissen: Requirements Information Management: The NATURE Approach

1994-25 * M. Jarke, K. Pohl, C. Rolland, J.-R. Schmitt: Experience-Based Method Evaluation and Improvement: A Process Modeling Approach

1994-26 * St. Jacobs, St. Kethers: Improving Communication and Decision Making within Quality Function Deployment

1994-27 * M. Jarke, H. W. Nissen, K. Pohl: Tool Integration in Evolving Information Systems Environments

1994-28 O. Burkart, D. Caucal, B. Steffen: An Elementary Bisimulation Decision Procedure for Arbitrary Context-Free Processes

1995-01 * Fachgruppe Informatik: Jahresbericht 1994

1995-02 Andy Schürr, Andreas J. Winter, Albert Zündorf: Graph Grammar Engineering with PROGRES

1995-03 Ludwig Staiger: A Tight Upper Bound on Kolmogorov Complexity by Hausdorff Dimension and Uniformly Optimal Prediction

1995-04 Birgitta König-Ries, Sven Helmer, Guido Moerkotte: An experimental study on the complexity of left-deep join ordering problems for cyclic queries

1995-05 Sophie Cluet, Guido Moerkotte: Efficient Evaluation of Aggregates on Bulk Types

1995-06 Sophie Cluet, Guido Moerkotte: Nested Queries in Object Bases

1995-07 Sophie Cluet, Guido Moerkotte: Query Optimization Techniques Exploiting Class Hierarchies

1995-08 Markus Mohnen: Efficient Compile-Time Garbage Collection for Arbitrary Data Structures

1995-09 Markus Mohnen: Functional Specification of Imperative Programs: An Alternative Point of View of Functional Languages

1995-10 Rainer Gallersdörfer, Matthias Nicola: Improving Performance in Replicated Databases through Relaxed Coherency

1995-11 * M.Staudt, K.von Thadden: Subsumption Checking in Knowledge Bases

1995-12 * G.V.Zemanek, H.W.Nissen, H.Hubert, M.Jarke: Requirements Analysis from Multiple Perspectives: Experiences with Conceptual Modeling Technology

1995-13 * M.Staudt, M.Jarke: Incremental Maintenance of Externally Materialized Views

1995-14 * P.Peters, P.Szczurko, M.Jeusfeld: Oriented Information Management: Conceptual Models at Work

1995-15 * Matthias Jarke, Sudha Ram (Hrsg.): WITS 95 Proceedings of the 5th Annual Workshop on Information Technologies and Systems

1995-16 * W.Hans, St.Winkler, F.Saenz: Distributed Execution in Functional Logic Programming

1996-01 * Jahresbericht 1995

1996-02 Michael Hanus, Christian Prehofer: Higher-Order Narrowing with Definitional Trees

1996-03 * W.Scheufele, G.Moerkotte: Optimal Ordering of Selections and Joins in Acyclic Queries with Expensive Predicates

1996-04 Klaus Pohl: PRO-ART: Enabling Requirements Pre-Traceability

1996-05 Klaus Pohl: Requirements Engineering: An Overview

1996-06 * M.Jarke, W.Marquardt: Design and Evaluation of Computer–Aided Process Modelling Tools

1996-07 Olaf Chitil: The Sigma-Semantics: A Comprehensive Semantics for Functional Programs

1996-08 * S.Sripada: On Entropy and the Limitations of the Second Law of Thermodynamics

1996-09 Michael Hanus (Ed.): Proceedings of the Poster Session of ALP96 - Fifth International Conference on Algebraic and Logic Programming

1996-09-0 Michael Hanus (Ed.): Proceedings of the Poster Session of ALP 96 - Fifth International Conference on Algebraic and Logic Programming: Introduction and table of contents

1996-09-1 Ilies Alouini: An Implementation of Conditional Concurrent Rewriting on Distributed Memory Machines

1996-09-2 Olivier Danvy, Karoline Malmkjær: On the Idempotence of the CPS Transformation

1996-09-3 Víctor M. Gulías, José L. Freire: Concurrent Programming in Haskell

1996-09-4 Sébastien Limet, Pierre Réty: On Decidability of Unifiability Modulo Rewrite Systems

1996-09-5 Alexandre Tessier: Declarative Debugging in Constraint Logic Programming

1996-10 Reidar Conradi, Bernhard Westfechtel: Version Models for Software Configuration Management

1996-11 * C.Weise, D.Lenzkes: A Fast Decision Algorithm for Timed Refinement

1996-12 * R.Dömges, K.Pohl, M.Jarke, B.Lohmann, W.Marquardt: PRO-ART/CE* — An Environment for Managing the Evolution of Chemical Process Simulation Models

1996-13 * K.Pohl, R.Klamma, K.Weidenhaupt, R.Dömges, P.Haumer, M.Jarke: A Framework for Process-Integrated Tools

1996-14 * R.Gallersdörfer, K.Klabunde, A.Stolz, M.Eßmajor: INDIA — Intelligent Networks as a Data Intensive Application, Final Project Report, June 1996

1996-15 * H.Schimpe, M.Staudt: VAREX: An Environment for Validating and Refining Rule Bases

1996-16 * M.Jarke, M.Gebhardt, S.Jacobs, H.Nissen: Conflict Analysis Across Heterogeneous Viewpoints: Formalization and Visualization

1996-17 Manfred A. Jeusfeld, Tung X. Bui: Decision Support Components on the Internet

1996-18 Manfred A. Jeusfeld, Mike Papazoglou: Information Brokering: Design, Search and Transformation

1996-19 * P.Peters, M.Jarke: Simulating the impact of information flows in networked organizations

1996-20 Matthias Jarke, Peter Peters, Manfred A. Jeusfeld: Model-driven planning and design of cooperative information systems

1996-21 * G.de Michelis, E.Dubois, M.Jarke, F.Matthes, J.Mylopoulos, K.Pohl, J.Schmidt, C.Woo, E.Yu: Cooperative information systems: a manifesto

1996-22 * S.Jacobs, M.Gebhardt, S.Kethers, W.Rzasa: Filling HTML forms simultaneously: CoWeb architecture and functionality

1996-23 * M.Gebhardt, S.Jacobs: Conflict Management in Design

1997-01 Michael Hanus, Frank Zartmann (eds.): Jahresbericht 1996

1997-02 Johannes Faassen: Using full parallel Boltzmann Machines for Optimization

1997-03 Andreas Winter, Andy Schürr: Modules and Updatable Graph Views for PROgrammed Graph REwriting Systems

1997-04 Markus Mohnen, Stefan Tobies: Implementing Context Patterns in the Glasgow Haskell Compiler

1997-05 * S.Gruner: Schemakorrespondenzaxiome unterstützen die paargrammatische Spezifikation inkrementeller Integrationswerkzeuge

1997-06 Matthias Nicola, Matthias Jarke: Design and Evaluation of Wireless Health Care Information Systems in Developing Countries

1997-07 Petra Hofstedt: Taskparallele Skelette für irregulär strukturierte Probleme in deklarativen Sprachen

1997-08 Dorothea Blostein, Andy Schürr: Computing with Graphs and Graph Rewriting

1997-09 Carl-Arndt Krapp, Bernhard Westfechtel: Feedback Handling in Dynamic Task Nets

1997-10 Matthias Nicola, Matthias Jarke: Integrating Replication and Communication in Performance Models of Distributed Databases

1997-11 * R. Klamma, P. Peters, M. Jarke: Workflow Support for Failure Management in Federated Organizations

1997-13 Markus Mohnen: Optimising the Memory Management of Higher-Order Functional Programs

1997-14 Roland Baumann: Client/Server Distribution in a Structure-Oriented Database Management System

1997-15 George Botorog: High-Level Parallel Programming and the Efficient Implementation of Numerical Algorithms

1998-01 * Fachgruppe Informatik: Jahresbericht 1997

1998-02 Stefan Gruner, Manfred Nagel, Andy Schürr: Fine-grained and Structure-Oriented Document Integration Tools are Needed for Development Processes

1998-03 Stefan Gruner: Einige Anmerkungen zur graphgrammatischen Spezifikation von Integrationswerkzeugen nach Westfechtel, Janning, Lefering und Schürr

1998-04 * O. Kubitz: Mobile Robots in Dynamic Environments

1998-05 Martin Leucker, Stephan Tobies: Truth - A Verification Platform for Distributed Systems

1998-06 * Matthias Oliver Berger: DECT in the Factory of the Future
1998-07 M. Arnold, M. Erdmann, M. Glinz, P. Haumer, R. Knoll, B. Paech, K. Pohl, J. Ryser, R. Studer, K. Weidenhaupt: Survey on the Scenario Use in Twelve Selected Industrial Projects
1998-08 * H. Aust: Sprachverstehen und Dialogmodellierung in natürlichsprachlichen Informationssystemen
1998-09 * Th. Lehmann: Geometrische Ausrichtung medizinischer Bilder am Beispiel intraoraler Radiographien
1998-10 * M. Nicola, M. Jarke: Performance Modeling of Distributed and Replicated Databases
1998-11 * Ansgar Schleicher, Bernhard Westfechtel, Dirk Jäger: Modeling Dynamic Software Processes in UML
1998-12 * W. Appelt, M. Jarke: Interoperable Tools for Cooperation Support using the World Wide Web
1998-13 Klaus Indermark: Semantik rekursiver Funktionsdefinitionen mit Striktheitsinformation
1999-01 * Jahresbericht 1998
1999-02 * F. Huch: Verifcation of Erlang Programs using Abstract Interpretation and Model Checking — Extended Version
1999-03 * R. Gallersdörfer, M. Jarke, M. Nicola: The ADR Replication Manager
1999-04 María Alpuente, Michael Hanus, Salvador Lucas, Germán Vidal: Specialization of Functional Logic Programs Based on Needed Narrowing
1999-05 * W. Thomas (Ed.): DLT 99 - Developments in Language Theory Fourth International Conference
1999-06 * Kai Jakobs, Klaus-Dieter Kleefeld: Informationssysteme für die angewandte historische Geographie
1999-07 Thomas Wilke: CTL+ is exponentially more succinct than CTL
1999-08 Oliver Matz: Dot-Depth and Monadic Quantifier Alternation over Pictures
2000-01 * Jahresbericht 1999
2000-02 Jens Vöge, Marcin Jurdzinski: A Discrete Strategy Improvement Algorithm for Solving Parity Games
2000-04 Andreas Becks, Stefan Sklorz, Matthias Jarke: Exploring the Semantic Structure of Technical Document Collections: A Cooperative Systems Approach
2000-05 Mareike Schoop: Cooperative Document Management
2000-06 Mareike Schoop, Christoph Quix (eds.): Proceedings of the Fifth International Workshop on the Language-Action Perspective on Communication Modelling
2000-07 * Markus Mohnen, Pieter Koopman (Eds.): Proceedings of the 12th International Workshop of Functional Languages
2000-08 Thomas Arts, Thomas Noll: Verifying Generic Erlang Client-Server Implementations
2001-01 * Jahresbericht 2000
2001-02 Benedikt Bollig, Martin Leucker: Deciding LTL over Mazurkiewicz Traces
2001-03 Thierry Cachat: The power of one-letter rational languages

2001-04	Benedikt Bollig, Martin Leucker, Michael Weber: Local Parallel Model Checking for the Alternation Free mu-Calculus
2001-05	Benedikt Bollig, Martin Leucker, Thomas Noll: Regular MSC Languages
2001-06	Achim Blumensath: Prefix-Recognisable Graphs and Monadic Second-Order Logic
2001-07	Martin Grohe, Stefan Wöhrle: An Existential Locality Theorem
2001-08	Mareike Schoop, James Taylor (eds.): Proceedings of the Sixth International Workshop on the Language-Action Perspective on Communication Modelling
2001-09	Thomas Arts, Jürgen Giesl: A collection of examples for termination of term rewriting using dependency pairs
2001-10	Achim Blumensath: Axiomatising Tree-interpretable Structures
2001-11	Klaus Indermark, Thomas Noll (eds.): Kolloquium Programmiersprachen und Grundlagen der Programmierung
2002-01 *	Jahresbericht 2001
2002-02	Jürgen Giesl, Aart Middeldorp: Transformation Techniques for Context-Sensitive Rewrite Systems
2002-03	Benedikt Bollig, Martin Leucker, Thomas Noll: Generalised Regular MSC Languages
2002-04	Jürgen Giesl, Aart Middeldorp: Innermost Termination of Context-Sensitive Rewriting
2002-05	Horst Lichter, Thomas von der Maßen, Thomas Weiler: Modelling Requirements and Architectures for Software Product Lines
2002-06	Henry N. Adorna: 3-Party Message Complexity is Better than 2-Party Ones for Proving Lower Bounds on the Size of Minimal Nondeterministic Finite Automata
2002-07	Jörg Dahmen: Invariant Image Object Recognition using Gaussian Mixture Densities
2002-08	Markus Mohnen: An Open Framework for Data-Flow Analysis in Java
2002-09	Markus Mohnen: Interfaces with Default Implementations in Java
2002-10	Martin Leucker: Logics for Mazurkiewicz traces
2002-11	Jürgen Giesl, Hans Zantema: Liveness in Rewriting
2003-01 *	Jahresbericht 2002
2003-02	Jürgen Giesl, René Thiemann: Size-Change Termination for Term Rewriting
2003-03	Jürgen Giesl, Deepak Kapur: Deciding Inductive Validity of Equations
2003-04	Jürgen Giesl, René Thiemann, Peter Schneider-Kamp, Stephan Falke: Improving Dependency Pairs
2003-05	Christof Löding, Philipp Rohde: Solving the Sabotage Game is PSPACE-hard
2003-06	Franz Josef Och: Statistical Machine Translation: From Single-Word Models to Alignment Templates
2003-07	Horst Lichter, Thomas von der Maßen, Alexander Nyßen, Thomas Weiler: Vergleich von Ansätzen zur Feature Modellierung bei der Softwareproduktlinienentwicklung
2003-08	Jürgen Giesl, René Thiemann, Peter Schneider-Kamp, Stephan Falke: Mechanizing Dependency Pairs
2004-01 *	Fachgruppe Informatik: Jahresbericht 2004

2004-02	Benedikt Bollig, Martin Leucker: Message-Passing Automata are expressively equivalent to EMSO logic
2004-03	Delia Kesner, Femke van Raamsdonk, Joe Wells (eds.): HOR 2004 – 2nd International Workshop on Higher-Order Rewriting
2004-04	Slim Abdennadher, Christophe Ringeissen (eds.): RULE 04 – Fifth International Workshop on Rule-Based Programming
2004-05	Herbert Kuchen (ed.): WFLP 04 – 13th International Workshop on Functional and (Constraint) Logic Programming
2004-06	Sergio Antoy, Yoshihito Toyama (eds.): WRS 04 – 4th International Workshop on Reduction Strategies in Rewriting and Programming
2004-07	Michael Codish, Aart Middeldorp (eds.): WST 04 – 7th International Workshop on Termination
2004-08	Klaus Indermark, Thomas Noll: Algebraic Correctness Proofs for Compiling Recursive Function Definitions with Strictness Information
2004-09	Joachim Kneis, Daniel Mölle, Stefan Richter, Peter Rossmanith: Parameterized Power Domination Complexity
2004-10	Zinaida Benenson, Felix C. Gärtner, Dogan Kesdogan: Secure Multi-Party Computation with Security Modules
2005-01 *	Fachgruppe Informatik: Jahresbericht 2005
2005-02	Maximillian Dornseif, Felix C. Gärtner, Thorsten Holz, Martin Mink: An Offensive Approach to Teaching Information Security: Aachen Summer School Applied IT Security
2005-03	Jürgen Giesl, René Thiemann, Peter Schneider-Kamp: Proving and Disproving Termination of Higher-Order Functions

* These reports are only available as a printed version.

Please contact biblio@informatik.rwth-aachen.de to obtain copies.