RWTH Aachen

Department of Computer Science
Technical Report

Satellites and Mirrors for Solving Independent Set on Sparse Graphs

Joachim Kneis and Alexander Langer

ISSN 0935–3232 · Aachener Informatik Berichte · AIB-2009-08

RWTH Aachen · Department of Computer Science · April 2009

Satellites and Mirrors for Solving Independent Set on Sparse Graphs*

Joachim Kneis, Alexander Langer

Dept. of Computer Science, RWTH Aachen University, Germany

Abstract. We study the well-known Maximum Independent Set (MIS) problem and introduce the notion of *satellites* of a node. Branching on nodes with satellites is extremely simple. Nevertheless, satellites can be used to overcome a couple of hard cases in previous algorithms. Together with the notion of *mirrors*, introduced by Fomin, Grandoni, and Kratsch, they can be used to solve MIS in time bounded by $O^*(1.1928^{m-n})$, which is $O^*(1.0922^n)$ for cubic graphs. This improves over previous results for sparse graphs.

1 Introduction

In this paper, we study the well-known Maximum Independent Set (MIS) problem: Given a graph $G = (V, E)$ with n nodes and m edges, the problem is to find an *independent set* $I \subseteq V$ of maximum size $\alpha(G)$, i.e., a set $I \subseteq V$ such that no two nodes in I are adjacent. This problem is known to be NP-complete [11] even on graphs of a maximum degree of three (*cubic graphs*). Being a problem with a long research history, there are already numerous results regarding approximation algorithms, randomized algorithms, or other approaches for MIS on sparse graphs, see, e.g., [1, 3, 4, 9]. Here, we concentrate on exact algorithms for sparse graphs. For exact algorithms on arbitrary graphs, we refer the reader to the latest corresponding results, which are due to Robson [13] and due to Fomin, Grandoni, and Kratsch [7].

Recently, there have been various new results for MIS on sparse graphs: In 1999, Beigel [2] introduced an $O^*(1.082^m)$ algorithm, which implies a run time bounded by $O^*(1.1259^n)$ for cubic graphs. This result was improved to $O^*(1.1254^n)$ by Chen, Kanj, and Xia [6]. Fürer [10] was the first to analyze the run time of algorithms for MIS in $m - n$, which eases the analysis of folding, an important reduction rule for nodes of degree two. He obtains a run time of $O^*(1.2365^{m-n})$, which is $O^*(1.1120^n)$ for cubic graphs. Subsequent improvements are due to Razgon [12] (to $O^*(1.1034^n)$), and only recently to $O^*(1.2048^{m-n})$, i.e., $O^*(1.0977^n)$ on cubic graphs, by Bourgeois, Escoffier, and Paschos [5]. In this paper, we improve the bound to $O^*(1.1928^{m-n})$, which is $O^*(1.0922^n)$ for cubic graphs.

While this improvement is of interest on its own, the new—to our best knowledge—notion of *satellites* can be used beyond the scope of this paper: firstly, if a node v has two adjacent satellites, then $\alpha(G) = \alpha(G[V \setminus \{v\}])$. Secondly, many exact algorithms for MIS use a bounded search tree technique and branch on some node v. Here, it is often disadvantageous if there are many edges in the neighborhood N of v, but few between N and the remaining graph. For example, four of the five worst case recurrences in the algorithm by Fomin,

* Supported by the DFG under grant RO 927/7

Grandoni, and Kratsch are graphs that contain satellites of v. These hard cases probably become easier when using our new technique. Analyzing the new run time, however, is very complex and subject to further research.

As we will demonstrate below, satellites can furthermore be combined efficiently with *mirrors*, a notation introduced by Fomin, Grandoni, and Kratsch [7], even though we cannot branch on mirrors and satellites simultaneously.

2 Preliminaries

Let $G = (V, E)$ be a graph. The set of nodes of G is denoted by $V(G)$. For a node $v \in V$, the (open) neighborhood of v is denoted by $N(v)$ and the closed neighborhood of v is denoted by $N[v] := N(v) \cup v$. Similarly, $N^2(v) := \bigcup_{u \in N(v)} N(u) \setminus N[v]$ and $N^2[v] := N^2(v) \cup N[v]$. This notation is extended to sets $U \subseteq V$ as follows: $N(U) := \bigcup_{u \in U} N(u) \setminus U$, $N[U] := N(U) \cup U$, $N^2(U) := \bigcup_{u \in U} N^2(u) \setminus N[U]$, and $N^2[U] := N^2(U) \cup N[U]$.

For $U \setminus V$, we by $G \setminus U$ denote the induced subgraph $G[V \setminus U]$ of V. As we will see later, trees play an important role for the run time analysis. We say G *contains* a tree, iff G has a maximal connected component that is a tree.

We now introduce *satellites*. Satellites allow for improved branching and let us introduce a new reduction rule.

Definition 1. *Let G be a graph $v \in V$. A node $u \in N^2(v)$ is called a* satellite *of v, if there is $u' \in N(v)$ such that $N[u'] \setminus N[v] = \{u\}$. In this case we also say u'* defines *u (as a satellite). The set of all satellites of a node v is denoted by $S(v) := \{\, u \in N^2(v) \mid u \text{ is a satellite of } v \,\}$.*

Lemma 1. *Let $G = (V, E)$ be a graph and $v \in V$. Then either $\alpha\big(G \setminus \{v\}\big) = \alpha(G)$, or $\alpha\big(G \setminus N[\{v\} \cup S(v)]\big) = \alpha(G) - |\{v\} \cup S(v)|$. If there are $u, w \in V$ such that $u, w \in S(v)$ and $\{u, w\} \in E$, then $\alpha(G) = \alpha\big(G \setminus \{v\}\big)$.*

Proof. Assume that every optimal independent set for G contains v. If there is a satellite u of v that is not contained in some optimal independent set I, we can replace v be the unique node in $N(v) \cap N(u)$ and obtain a new independent set of equal size, a contradiction. If v has two adjacent satellites u and w, $\alpha(G) = \alpha\big(G \setminus \{v\}\big)$ is concluded immediately. □

We also use *mirrors*, a notion introduced by Fomin, Grandoni, and Kratsch [7].

Definition 2. *Let G be a graph $v \in V$. A node $u \in N^2(v)$ is called a* mirror *of v if $N(v) \setminus N(u)$ is a clique. We let $M(v) := \{\, u \in N^2(v) \mid u \text{ is a mirror of } v \,\}$.*

Lemma 2 (Fomin, Grandoni, Kratsch [7]). *Let $G = (V, E)$ be a graph, $v \in V$ and let I be an independent set of size k in G that does not contain v. Then $G \setminus (\{v\} \cup M(v))$ contains in independent set of size k as well.*

Since the Measure & Conquer technique [8] can often be used to improve the run time bounds, we do not simply measure the run time of our analysis in $m - n$, but allow edges and nodes to weighted differently. Although it turns out that $\lambda = \mu = 1$ is optimal for the branching vectors obtained in this paper, future improvements for some of our branches might lead to other better optimal values of λ and μ, which is why we keep λ and μ as variables.

Definition 3. *Let $G = (V, E)$ be graph and let $\lambda, \mu \in \mathbf{R}^+$, such that $\lambda \geq \mu$. We call $\varphi(G) := \lambda \cdot |E| - \mu \cdot |V|$ the* measure *of G.*

We now introduce the reductions rules applied by our algorithm. The first trivial reduction rule is to remove isolated nodes. Note that this reduction rule increases our measure by μ, hence we must be careful whenever isolated nodes appear. We will later show that all reduction rules only increase the measure when applied to maximal components that are trees.

Definition 4. *Let $G = (V, E)$ be a graph. For two adjacent nodes $u, v \in V$, we say u* dominates *v iff $N[u] \supseteq N[v]$. If $v \in V$ is a node of degree two, let u_1, u_2 be the neighbors of v in G and $N := N(u_2) \setminus \{v\}$. The operation of adding the edges $\{v', u_1\}$ for all $v' \in N$ to G and removing v and u_2 from G is called* folding.

Note that dominating nodes can easily removed from the graph, as this does not change the size of an optimal independent set. In particular, if a graph contains a node of degree one, its neighbor is removed.

Folding is a well-known reduction rule and can be applied to all nodes of degree two that are not part of triangle. However, nodes of degree two that are contained in a triangle are dominated by both neighbors. Thus, our graphs never contain nodes of degree two or less after the reduction rules have been applied. It is easy to see that folding does not increase the measure.

Lemma 3 (Fürer [10]). *Let $G = (V, E)$ be a connected graph, and let $v, u \in V$ such that $G \setminus \{u, v\}$ consists of two components G_1, G_2 that are not connected to each other. If $\varphi(G_1) \leq c$ for some fixed c, there is a graph G' such that $V(G) \subseteq V(G_2) \cup \{v, u\}$, G_2 is an induced subgraph of G' and $\alpha(G') = \alpha(G) + k$ and $\varphi(G') \leq \varphi(G)$, where k depends only on G_1.*

We can therefore assume that our graphs never contain separators of size at most two: either one component is small and we apply Fürer's reduction rule, or all components are big and branching on the separator is efficient, since the graph decomposes into big connected components, each of which contributes to the running time only additively. Furthermore, if there are no small separators, we do not end up with a forest when a set of nodes is removed from the graph, i.e., G contains at least one maximal connected component that is not a tree. This will be useful in the analysis.

The last reduction rule is to remove induced cycles of length four whose nodes are all of degree three (*cubic cycles*). Again, this reduction rule eases our analysis. Note that if such an cycle is not induced, i.e., it contains a chord, one of its nodes is removed because of domination. See Figure 1 for an example.

Lemma 4. *Let G be a graph that contains an induced cycle (u_1, u_2, u_3, u_4) of length four, such that $\deg(u_i) = 3$ for all $1 \leq i \leq 4$. Let G' be the graph obtained from G by removing $\{u_1, \ldots, u_4\}$ and adding edges for the cycle (u'_1, u'_2, u'_3, u'_4) where $\{u'_i\} = N(u_i) \setminus \{u_1, u_2, u_3, u_4\}$ and they do not yet exist. Then $\alpha(G) = \alpha(G') + 2$ and $\varphi(G) \geq \varphi(G')$.*

Proof. Let $U' := \{u'_1, u'_2, u'_3, u'_4\}$ (see Figure 1). By a simple exchange argument, first note that there is an optimal independent set I in G that contains two nodes from $U := \{u_1, u_2, u_3, u_4\}$. Wlog, assume $\{u_1, u_3\} \subseteq I$, and therefore

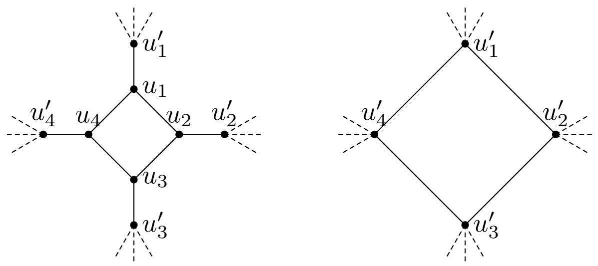

Fig. 1. Reduction rule for cubic cycles of length four, such that $\alpha(G) = \alpha(G') + 2$.

Algorithm 1 Input: A graph $G = (V, E)$, Output: $\alpha(G)$

Reduce G according to the reduction rules;
If G contains $t > 1$ maximal connected components $G_1, \ldots, G_t$, then
    return $\alpha(G_1) + \cdots + \alpha(G_t)$;
Pick $v \in V$ such that branching on v yields the best branching vector;
If the mirror branch of v yields a better branching vector, then
    return $\max\big(\alpha\big(G \setminus (\{v\} \cup M(v))\big), \alpha\big(G \setminus N[v]\big)+1\big)$;
else
    return $\max\big(\alpha\big(G \setminus \{v\}\big), \alpha\big(G \setminus N[v \cup S(v)]\big) + 1 + |S(v)|\big)$;

$I \cap \{u_1', u_2', u_3', u_4'\} \subseteq \{u_2', u_4'\}$. Since $I' = I \setminus U$ is an independent set for G', $\alpha(G) \leq \alpha(G') + 2$.

Similarly, any optimal independent set I' in G' either has $I' \cap U' \subseteq \{u_1', u_3'\}$ or $I' \cap U' \subseteq \{u_2', u_4'\}$. Wlog, assume $I' \cap U' \subseteq \{u_1', u_3'\}$. Then $I' \cup \{u_2, u_4\}$ is an independent set for G, implying $\alpha(G') + 2 \leq \alpha(G)$.

Let m' be the number of edges in $G[U']$. Then $\varphi(G) = m\lambda - n\mu \geq (m - 4 - m')\lambda - (n - 4)\mu = \varphi(G')$. □

Definition 5. *Let $G = (V, E)$. We call the graph $R(G)$ obtained from G by applying the reduction rules above, i.e., removing nodes with adjacent satellites, removing dominating nodes, applying folding and Fürer's reduction rule as well as applying Lemma 4 until no further reduction rules can be applied, the* reduced graph *of G. A graph G is called* reduced, *if $G = R(G)$.*

For $U \subseteq V$, the measure difference between G and $R(G \setminus U)$ is defined as

$$\Delta_\varphi(U) := \varphi(G) - \varphi(R(G \setminus U)).$$

We can now combine all reduction rules as well as our branching rules for mirrors and satellites into Algorithm 1. Note that we do not branch simultaneously on mirrors and satellites, as there are graphs where this does not result in a correct solution. For readability, we simply branch on the node that yields the best branching vector, i.e., guarantees the best run time. However, a more efficient strategy to select v can easily obtained, when using the same nodes as in the proofs of Lemmata 7, 8, and 9. The correctness of Algorithm 1 easily follows from the lemmata above.

Lemma 5. *Let $G = (V, E)$ be a graph that does not contain a tree and let $U \subseteq V$. Then, $\Delta_\varphi(U) \geq 0$.*

Proof. All reduction rules except removing isolated nodes do not increase the measure $\varphi = \varphi(G)$, since always more edges than nodes are removed from G. Furthermore, only removing nodes in U or removing nodes because of domination can increase the number of maximal connected components.

Firstly, we iteratively remove the nodes $u \in U$ and each iteration decreases the measure φ by $\lambda \deg(u) - \mu \geq \mu(\deg(u) - 1)$ (note that particularly φ is increased by μ when isolated nodes are removed). Furthermore, when u is removed, the number of trees is increased by at most $\deg(u) - 1$: A (maximal) connected component of G decomposes into at most $deg(u)$ new components, but this maximum occurs only if the component that contains u itself already is a tree. If u is an isolated node, φ increases by μ, while t decreases by one. Hence, $G \setminus U$ contains at most $t \leq (\varphi(G) - \varphi(G \setminus U))/\mu$ trees.

$G \setminus U$ contains t trees and since applying the reduction rules for trees eliminates these trees, the measure increases by at most μ for each tree created by the removal of U. An analogous argument holds for all trees created by the domination reduction rule.

Therefore,

$$\varphi(G \setminus U) \geq \varphi(R(G \setminus U)) - \mu t$$

and

$$\varphi(G) \geq \mu t + \varphi(G \setminus U) \geq \mu t + \varphi(R(G \setminus U)) - \mu t = \varphi(R(G \setminus (U))).$$

□

Setting $U := \emptyset$, this lemma implies that the reduction rules do not increase the measure of graphs without trees. If G is a tree, $R(G)$ is the empty graph, and thus $\varphi(G) - \varphi(R(G)) \geq -\mu$. Hence, if G contains t trees, removing these trees increases the measure by $t\mu$. In the analysis of our algorithm, it is therefore sufficient to estimate an increase of at most μ in the measure whenever a tree is created.

3 Branching and Analysis

In this section, we study the change of the measure $\varphi(G)$ when an algorithm branches on the two cases $G \setminus \{v\}$ (v is not contained in an optimal independent set) and $G \setminus N[v]$ (v is contained in the solution). The actual branching in Algorithm 1 uses all information given in the previous section, including the rules for mirrors and satellites. Unfortunately, some branches turn out to be insufficiently efficient. However, in these cases we can guarantee the existence of a node of degree four, which then allows for a better combined branch.

Lemma 6. *Let G be a reduced graph of maximum degree $d \geq 5$ and $v \in V$ such that $\deg(v) = d$. Then branching as described in Algorithm 1 either yields a branching vector at least as good as $(5\lambda - \mu, 13\lambda - 6\mu)$ or $(6\lambda - \mu, 13\lambda - 7\mu)$, or yields at least the branching vector $(5\lambda - \mu, 13\lambda - 7\mu)$, but $R(G \setminus \{v\})$ contains a node of degree at least four.*

Proof. For $\lambda = \mu = 1$, a similar result can already be found in [5]. We give a more detailed proof for arbitrary $\lambda \geq \mu$.

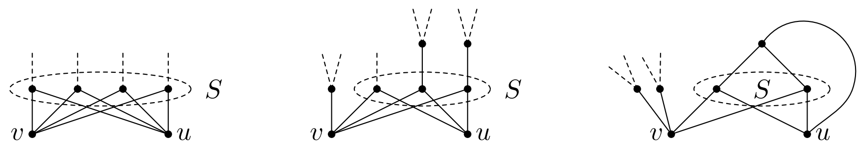

Fig. 2. Branching on graphs with mirrors. Dashed edges are known to exist, although their endpoints are unknown. S contains neighbors of degree three shared by v and its mirror u. The case $|S| = 4$ is depicted on the left, the second picture shows $|S| = 3$ and $deg(u) = 3$. For $|S| = 2$, the right picture is an example why no path of length three can exist in $G \setminus \{u, v\}$ (domination by u).

Note that $G \setminus \{v\}$ does not contain a tree since there are no nodes of degree one. Therefore, $\Delta_\varphi(v) \geq d\lambda - \mu$. If $G \setminus N[v]$ does not contain a tree, $\Delta_\varphi(N[v]) \geq 13\lambda - 6\mu$: each node in $N(v)$ is of degree at least three and since no node in $N(v)$ is dominated, each $u \in N(v)$ has at least one neighbor in $N^2(v)$.

Now assume $G \setminus N[v]$ contains t trees. Since there are at least three edges between $N(v)$ and each tree, and there are at least three edges from $N(v)$ to nodes in the remaining graph (no small separators), and since each node in $N(v)$ has at least one neighbor in $N^2(v)$ (domination), there are at least $3t + 3 \geq d$ edges between $N(v)$ and $N^2(v)$. Moreover, there are d edges between v and $N(v)$ and thus at least

$$\begin{aligned} & 3t + 3 + d + \left\lceil \frac{\left(\sum_{u \in N(v)} \deg(u)\right) - (3t + 3 + d)}{2} \right\rceil \\ \geq \quad & 3t + 3 + d + \left\lceil \frac{3d - (3t + 3 + d)}{2} \right\rceil =: \Delta_m(t, d) \end{aligned}$$

edges incident to $N[v]$, because each node is of degree at least three and only edges in $G[N(v)]$ might be counted twice. Thus, $\Delta_\varphi(N[v]) \geq \Delta_m(t, d)\lambda - (d + t + 1)\mu$.

If $t \geq 2$ or $d \geq 6$, we already have $\Delta_\varphi(N[v]) \geq 15\lambda - 8\mu$. If $t = 1$, $d = 5$, and $N(v)$ contains at least one node of degree four, we obtain $\Delta_\varphi(N[v]) \geq 14\lambda - 6\mu - \mu$. However, if $t = 1$, $d = 5$, and all nodes in $N(v)$ are of degree three, we only obtain $\Delta_\varphi(N[v]) \geq 13\lambda - 6\mu - \mu$. Fortunately, in this case, $R(G \setminus \{v\})$ either contains a node of degree four because of folding, or we gain an additional edge because of folding or domination, and therefore $\Delta_\varphi(v) \geq (5 + 1)\lambda - \mu$. □

Lemma 7. *Let $G = (V, E)$ be a reduced graph of maximum degree four, $v \in V$ such that $M(v) \neq \emptyset$ and $\deg(v) = 4$. Then branching as described in Algorithm 1 yields at least the branching vector $(7\lambda - 2\mu, 10\lambda - 5\mu)$.*

Proof. Let $u \in M(v)$ and let $S = \{ u' \in N(v) \mid u' \in N(u), \deg(u') = 3 \}$. Then, a node w has degree one in $G \setminus \{v, u\}$ if and only if $w \in S$. Note that by domination all nodes in $N[S]$ are removed in $R(G \setminus \{v, u\})$. Moreover, $G[S]$ does not contain an edge $\{w, w'\}$, because otherwise w and w' dominated each other.

If $|S| = 4$, $S = N(v)$ implies that $N^2[v]$ is completely removed in $R(G \setminus \{v, u\})$. Thus, we gain twelve edges and six nodes when removing $N[v] \cup \{u\}$. Since $|N^2(v) \setminus \{u\}| \geq 3$ (no small separators), and since each $w \in N(v)$ is of degree three and connected to both u and v, $G \setminus N[v] \cup \{u\}$ does not contain a tree. Therefore, removing all remaining nodes in $N^2(v) \setminus \{u\}$ and applying the reduction rules afterwards cannot increase the measure by Lemma 5 and hence $\Delta_\varphi(\{v,u\}) \geq 12\lambda - 6\mu$.

If $|S| = 3$ and $\deg(u) = 4$, $G \setminus \{v,u\}$ contains at most one tree as there are only three nodes of degree one and each tree has at least two leafs. Thus $\Delta_\varphi(\{v,u\}) \geq 8\lambda - 2\mu - \mu$, since we remove eight edges and two nodes.

If $|S| = 3$ and $\deg(u) = 3$, at least two nodes in S must be connected to different nodes in $N^2(v) \setminus \{u\}$, i.e., also not be connected to the remaining node in $N(v) \setminus S$ (otherwise G contains a separator of size two). See Figure 2 for an example. But then $G \setminus \{v,u\}$ contains no tree: at least two nodes of degree one are connected to different nodes of degree three. Since there are only three nodes of degree one in $G \setminus \{v,u\}$ at all, this subgraph cannot be a tree. Therefore, we remove seven edges and two nodes and gain $\Delta_\varphi(\{v,u\}) \geq 7\lambda - 2\mu$.

If $S = \{u_1, u_2\}$, every tree contained in $G\setminus\{v,u\}$ must be path, because there are only two nodes of degree one (namely u_1 and u_2). Assume, there is such a path P. Since u_1 and u_2 are not connected, P must contain at least one additional node, which then has degree two. Removing v and u only influences the degree of nodes in $N(\{v,u\})$, hence all nodes in $V \setminus N[\{v,u\}]$ are still of degree at least three in $G\setminus\{v,u\}$. Therefore, $P \subseteq N(\{v,u\})$. Similarly, $N[w] \subseteq N(\{v,u\})$ for all nodes $w \in P$, because if some $w \in P$ has a neighbor w' in $V \setminus N[v,u]$, then P is not a path in $G\setminus\{v,u\}$ ($\deg(w') \geq 3$). If a node $w \in P$ is adjacent to both u_1 and u_2, i.e., P contains three nodes, w is dominated by v or u (see Figure 2, where domination by u is depicted). If P contains at least four nodes, $(N(v) \cup N(u)) \setminus P$ is a separator of size at most two, because $|N(v) \cup N(u)| \leq 6$. Thus, $G \setminus \{v,u\}$ contains no tree.

Finally, if $|S| = 1$, then $G \setminus \{v,u\}$ cannot contain a tree either (only one node has degree one) and thus we obtain $\Delta_\varphi(\{v,u\}) \geq 7\lambda - 2\mu$ for $|S| \leq 2$ (four edges for v, at least three edges for u).

For the second component of the branching vector, $\Delta_\varphi(N[v]) \geq 10\lambda - 5\mu$ is easily obtained, if $G\setminus N[v]$ contains no tree. If $G\setminus N[v]$ contains t trees, there must be at least $3t$ edges between $N(v)$ and these trees, at least three edges from $N(v)$ to further nodes since there are no small separators, and of course there are four edges between v and $N(v)$. Therefore, if $t \geq 2$, $\Delta_\varphi(N[v]) \geq (3t+7)\lambda - (5+t)\mu \geq 13\lambda - 7\mu$. If $t = 1$, there must be at least one additional edge incident to $N(v)$, because the minimum degree is three. Hence, $\Delta_\varphi(N[v]) \geq (3+7+1)\lambda - (5+1)\mu = 11\lambda - 6\mu$. This argument is similar to the one given by Bourgeois, Escoffier, and Paschos [5] for $\lambda = \mu = 1$.

Overall, we obtain at least the branching vector $(7\lambda - 2\mu, 10\lambda - 5\mu)$. □

Lemma 8. *Let $G = (V,E)$ be a reduced graph of maximum degree four, $v \in V$ such that* $\deg(v) = 4$*, $M(v) = \emptyset$, and $S(v) \neq \emptyset$. Then branching as described in Algorithm 1 yields at least the branching vector* $(9\lambda - 4\mu, 7\lambda - 2\mu)$.

Proof. Let $u \in S(v)$ and let $\{w\} \in N(u) \cap N(v)$ define u. Since the only neighbor of w in $N^2(v)$ is u, v dominates w in $G \setminus \{u\}$. Similarly to the two cases $|S| \leq 2$ in the proof of Lemma 7, $G \setminus \{v,u\}$ does not contain a tree (the argument given there did not use the mirror property of u). Thus, $\Delta_\varphi(u) \geq 7\lambda - 2\mu$, as at least seven edges and exactly two nodes are removed.

If $\deg(u) = 4$, we obtain $\Delta_\varphi(N[u]) \geq 10\lambda - 5\mu$, again as in the proof of Lemma 7. Thus, we can now assume $\deg(u) = 3$.

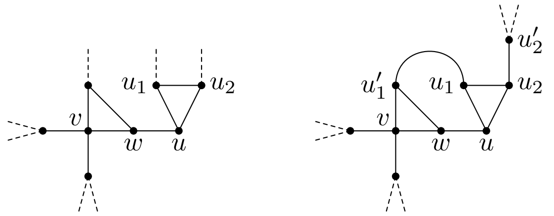

Fig. 3. A satellite u of v on a triangle. Existence of dashed edges is given by a minimum degree of three per node, although their endpoints are unknown. a) If $s := \deg(w)+\deg(u_1)+\deg(u_2) > 9$, then $\Delta_\varphi(N[u]) \geq 9\lambda - 4\mu$. b) If $s = 9$, removing $N[u] \cup \{u_1', u_2'\}$ affects at least eleven edges. A possible lower bound is shown.

If $G \setminus N[u]$ contains a tree, this tree is not an isolated node x: Assume x is an isolated node. Then due to degree reasons, x must be adjacent to all nodes in $N(u)$, and one of these nodes is w. Since w defines u as a satellite, $x \in N[v]$. If $x \neq v$, x is not an isolated node in $G \setminus N[u]$, since x is still adjacent to v. If $x = v$, then $|N(v) \cap N(u)| = 3$, i.e., $u \in M(v)$, a contradiction. Hence, if $G \setminus N[u]$ contains a tree, then there are at least four edges from $N(u)$ to the tree. Since there are no separators of size at most two, there are at least ten edges incident to $N[u]$. Similarly, the number of trees in $G \setminus N[u]$ can be bounded by one, since there are at most nine edges between $N(u)$ and $N^2(u)$. Therefore $\Delta_\varphi(N[u]) \geq 10\lambda - 4\mu - \mu$ when $G \setminus N[u]$ contains a tree.

Hence, we now assume $G \setminus N[u]$ does not contain a tree. If $N(u)$ contains at least two nodes of degree four, we easily obtain $\Delta_\varphi(N[u]) \geq 9\lambda - 4\mu$. Similarly, if u is not part of a triangle, there are at least nine edges incident to $N[u]$ and hence $\Delta_\varphi(N[u]) \geq 9\lambda - 4\mu$.

Finally, assume that u is part of some triangle. Note that this implies $N[w] \cap N(u) = \{w\}$: If there is a node $w' \in N(u) \cap N(w)$, either w dominates w' or w' dominates w, since at most one node in $N(u)$ is of degree four. Thus, u is contained in exactly the triangle (u, u_1, u_2) where $N(u) \setminus N[w] = \{u_1, u_2\}$. For an example, see Figure 3. If $\deg(u') = 4$ for some $u' \in N(u)$, this implies again $\Delta_\varphi(N[u]) \geq 9\lambda - 4\mu$.

Hence, we can assume $\deg(u') = 3$ for all $u' \in N[u]$. Since $\deg(u_1) = \deg(u_2) = 3$, there is a $u_1' \in N(u_1) \setminus N(u_2)$ (and similarly u_2'). Otherwise, u_1 is dominated by u_2. Note that $u_1', u_2' \in S(u)$, and thus, both are not adjacent. Since $\deg(w) = 3$, $|N(w) \cap \{u_1', u_2'\}| \leq 1$, because the two nodes on $N(w) \setminus \{u\}$ are connected, i.e., $N(w) \setminus \{u\} \subseteq N[v]$ and $v \in N(w) \setminus \{u\}$ (recall that w defines u). This situation is exemplified in Figure 3, under the assumption that $u_1' \in N(w)$.

Therefore, there are at least eleven edges incident to $N[u] \cup \{u_1', u_2'\}$: four edges in $G[N[u]]$, four edges between $N(u)$ and $N^2(u)$ because $G[N[u]]$ contains exactly one triangle, and three other edges incident to $\{u_1', u_2'\}$ (four if $|N(w) \cap \{u_1', u_2'\}| = 0$).

Recall now that u_1' and u_2' are satellites of u and we therefore can branch on $G \setminus (N[u] \cup \{u_1', u_2'\})$: If $G \setminus (N[u] \cup \{u_1', u_2'\})$ contains no tree, we thus obtain $\Delta_\varphi(N[\{u, u_1', u_2'\}]) \geq 11\lambda - 6\mu$ by Lemma 5. If $G \setminus (N[u] \cup \{u_1', u_2'\})$ contains t trees, there at least $3t+3$ edges between $N[u] \cup \{u_1', u_2'\}$ and $V \setminus (N[u] \cup \{u_1', u_2'\})$ and thus $\Delta_\varphi(N[\{u, u_1', u_2'\}]) \geq (3t + 3 + 6)\lambda - 6\mu - t\mu \geq 12\lambda - 7\mu$, by applying Lemma 5 to each component of $G \setminus N[u] \cup \{u_1', u_2'\}$ that is not a tree.

Overall, we obtain at least the branching vector $(9\lambda - 4\mu, 7\lambda - 2\mu)$. $\square$

Lemma 9. *Let $G = (V, E)$ be a reduced graph of maximum degree four, and let $v \in V$ such that $M(v) = \emptyset$ and $S(v) = \emptyset$. Then branching as described in Algorithm 1 either yields a branching vector at least as good as $(5\lambda - \mu, 12\lambda - 5\mu)$, $(4\lambda - \mu, 13\lambda - 5\mu)$, or $(7\lambda - 2\mu, 9\lambda - 4\mu)$, or yields a branching vector at least as good as $(4\lambda - \mu, 12\lambda - 5\mu)$, but $R(G \setminus \{v\})$ contains a node of degree four.*

Proof. $M(v) = \emptyset$ implies that each node in $N^2(v)$ has at most two neighbors in $N(v)$, while $S(v) = \emptyset$ implies that each node in $N(v)$ has at least two neighbors in $N^2(v)$.

If all nodes in $N(v)$ are of degree four, we obtain $\Delta_\varphi(v) \geq 4\lambda - \mu$. If $N(v)$ contains at least one node of degree three, either $\Delta_\varphi(v) \geq 5\lambda - \mu$, or $\Delta_\varphi(v) \geq 4\lambda - \mu$, but $R(G \setminus \{v\})$ again contains a node of degree at least four.

Since there are at most twelve edges between $N(v)$ and $N^2(v)$ and since each node is of degree at least three, $G \setminus N[v]$ contains at most six nodes of degree one (there are none of degree zero, because otherwise $M(v) \neq \emptyset$). Now if $G \setminus N[v]$ contains three trees, each of this trees consists of a single edge between two of these nodes only. Hence, G either is of constant size or is not connected.

If $G \setminus N[v]$ contains two trees T_1 and T_2, each of these trees contains at least two nodes of degree one, which is only possible if there are eight edges between $T_1 \cup T_2$ and $N(v)$. Since G does not contain a separator of size two, $N(v)$ must contain at least three nodes that are connected to three nodes in $N^2(v) \setminus (V(T_1) \cup V(T_2))$. But then, there are least 15 edges incident to nodes in $N[v]$ and thus $\Delta_\varphi(N[v]) \geq 15\lambda - 5\mu - 2\mu$.

If $G \setminus N[v]$ contains no tree, we immediately obtain $\Delta_\varphi(N[v]) \geq 13\lambda - 5\mu$, if at least one node in $N(v)$ is of degree four. Otherwise, we obtain $\Delta_\varphi(N[v]) \geq 12\lambda - 5\mu$.

Let us therefore now assume that $G \setminus N[v]$ contains exactly one tree. If all nodes in $N(v)$ are of degree four, we have $\Delta_\varphi(N[v]) \geq 14\lambda - 5\mu - \mu$. If $N(v)$ contains at least one node of degree four, there are at least 13 edges incident to $N[v]$, because each node in $N(v)$ has at least two neighbors in $N^2(v)$. Thus, we have $\Delta_\varphi(N[v]) \geq 13\lambda - 6\mu$.

If $\deg(u) = 3$ for all $u \in N(v)$ and $G \setminus N[v]$ contains a tree, there are at most five edges between $N(v)$ and this tree. Therefore, the tree must be an isolated edge $\{w_1, w_2\}$ or a path (w_1, w', w_2) of length two (and $\deg(w_1) = \deg(w_2) = 3$ or $\deg(w_1) = \deg(w_2) = \deg(w') = 3$ in G, respectively). Otherwise, there are at least six edges between $N(v)$ and the tree which implies that $N^2(v)$ contains at most two nodes that are not part of the tree, which yields a separator of size two. Moreover, w_1 and w_2 share at most one neighbor in $N(v)$, because otherwise $N(v) \setminus (N(w_1) \cup N(w_2))$ is again a separator of size two.

Assume $G \setminus N[v]$ contains a path (w_1, w', w_2) such that $\deg(w_1) = \deg(w_2) = \deg(w') = 3$. If $N(w_1) \cap N(w') \neq \emptyset$, then $|N(w_1) \cap N(w')| = 1$ because of the domination rule, and both v and the unique node $u' \in N(w_1) \setminus N[w']$ are satellites of w' and connected. This situation is depicted in Figure 4. Thus, wlog $N(w_i) \cap N(w') = \emptyset$. But then, $N(w_1) \cap N(w_2) \setminus \{w'\} \neq \emptyset$ since $|N(v)| = 4$, and thus $|(N(w_1) \cup N(w_2)) \setminus \{w'\}| = 3$. Let $\{u'\} = (N(w_1) \cap N(w_2)) \setminus \{w'\}$. Then, (w_1, w', w_2, u') is an induced cycle of length four, all whose nodes are of degree three. Thus, G is not reduced by Lemma 4.

Hence, if $G \setminus N[v]$ contains a tree, this tree must be a single edge $\{w_1, w_2\}$, and there are exactly four edges between $\{w_1, w_2\}$ and $N(v)$, and thus also four edges

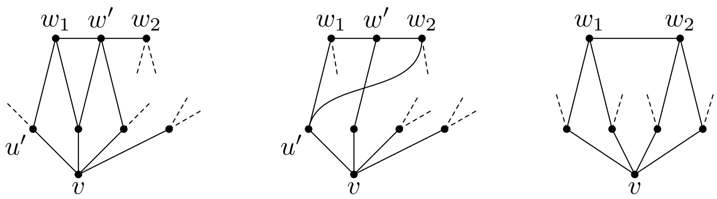

Fig. 4. Some cases we study when branching on v with $M(v) = S(v) = \emptyset$. If $G \setminus N[v]$ contains a path (w_1, w', w_2) as depicted on the left, v and u' are adjacent satellites of w'. If w' has a neighbor out of $N(\{w_1, w_2\})$ as depicted in the middle, then w_1 and w_2 have at least one common neighbor u' and thus an induced cubic cycle of length four exists. If $G \setminus N[v]$ contains an isolated edge (w_1, w_2) as depicted on the right, then $v \in M(w_1)$ and we branch on w_1 instead.

between $N(v)$ and the remaining graph. Therefore, $\Delta_\varphi(N[v]) \geq 12\lambda - 5\mu - \mu$. However, in this case, either $\Delta_\varphi(v)$ is larger or branching on w_1 gives a better result:

If $|(N(w_1) \cup N(w_2)) \cap N(v)| = 3$, i.e., one neighbor is shared, both w_1 and w_2 dominate the unique node $w' \in N(w_1) \cap N(w_2)$, after v is removed. Thus four edges incident to v and five edges incident to $\{w_1, w_2\}$ are removed and $\Delta_\varphi(v) \geq 9\lambda - 4\mu$. Moreover, four nodes are removed (v, w_1, w_2, w'). The remaining graph does not contain a tree, since the three remaining nodes in $N(v)$ must be connected to three different nodes in $N^2(v)$. This yields the branching vector $(12\lambda - 6\mu, 9\lambda - 4\mu)$.

If $|N(w_1) \cup N(w_2)| = 4$, v is a mirror of w_1 (see Figure 4). We now branch on w_1 instead: $\Delta_\varphi(\{w_1, v\}) \geq 7\lambda - 2\mu$ and $G \setminus \{w_1, v\}$ does not contain a tree (the only two nodes of degree one have neighbors of degree at least three). Similarly, $G \setminus N[w_1]$ does not contain a tree, since there are no nodes of degree one at all. Therefore, $\Delta_\varphi(N[w_1]) \geq 9\lambda - 4\mu$.

In summary, if the first branch does not lead to a node of degree at least four, we obtain branching vectors as least as good as either $(5\lambda - \mu, 12\lambda - 5\mu)$, $(4\lambda - \mu, 13\lambda - 5\mu)$, or $(7\lambda - 2\mu, 9\lambda - 4\mu)$. If otherwise the first branch yields at least one node of degree at least four, we obtain a branching vector at least as good as $(4\lambda - \mu, 12\lambda - 5\mu)$. $\square$

Lemma 10. *Let $G = (V, E)$ be a reduced graph of maximum degree three and let $v \in V$ be such that v is on a triangle (v, u_1, v_2). Then branching as described in Algorithm 1 either yields a branching vector at least as good as $(10\lambda - 6\mu, 12\lambda - 7\mu)$, or yields a branching vector at least as good as $(7\lambda - 5\mu, 12\lambda - 7\mu)$, but $R(G \setminus \{v\})$ contains a node of degree at least four.*

Proof. Let v', u_1', u_2' be the unique neighbors of v, u_1, and u_2, respectively, that are not in $\{v, u_1, u_2\}$. First note that $|\{v', u_1', u_2'\}| = 3$, since otherwise at least two nodes in $\{v, u_1, u_2\}$ dominate each other. This also means that v' is a satellite of u_1 and u_2, u_1' is a satellite of v and u_2, as well as u_2' is a satellite of v and u_1.

Furthermore, if either two of $\{v', u_1', u_2'\}$ are connected, there are connected satellites and there is no need to branch. For symmetry reasons, we can therefore assume that there is no edge in $G[\{v', u_1', u_2'\}]$. This situation is depicted in Figure 5.

$G \setminus \{v\}$ does not contain a tree (no nodes of degree one). If v_1' and v_2', the other neighbors of v', are adjacent, we have $\Delta_\varphi(v) \geq 10\lambda - 6\mu$, else $\Delta_\varphi(v) \geq 7\lambda - 5\mu$, but we obtain a node of degree four.

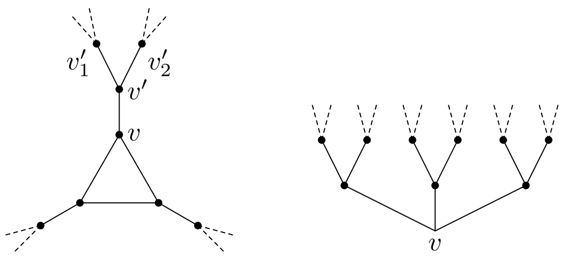

Fig. 5. Branching on a cubic graph G. If G contains triangles, by the reductions rules for domination and satellites we have the situation depicted on the left. Otherwise, G does neither contain cycles of length three or four, and the situation is as depicted on the right. Again, dashed edges are known to exist, although their endpoints are unknown.

$G - N[v] - S(v)$ contains at most one tree, otherwise there is a separator of size at most two in G. Counting only the edges in $G[N[v \cup S(v)]$ and the six nodes in $N[v] \cup S(v)$, it therefore is $\Delta_\varphi(N[v \cup S(v)]) \geq 12\lambda - 6\mu - \mu$. □

Lemma 11. *Let $G = (V, E)$ be a reduced, triangle-free graph of maximum degree three, and let $v \in V$. Then branching as described in Algorithm 1 either yields a branching vector at least as good as $(5\lambda - \mu, 9\lambda - 4\mu)$, or yields a branching vector at least as good as $(3\lambda - \mu, 9\lambda - 4\mu)$, but $R(G \setminus \{v\})$ contains a node of degree at least four.*

Proof. Since G is reduced and triangle-free, it does neither contain cycles of length three nor of length four. Therefore, $|N^2(v)| = 6$. This situation is depicted in Figure 5. Since G does not contain cycles of length four, $G \setminus \{v\}$ contains exactly three nodes of degree two, namely $N(v)$. Let $N(v) = \{u_1, u_2, u_3\}$. Since $|N^2(v)| = 6$, folding u_1 does not create an edge in $N(u_2)$ nor in $N(u_3)$. Therefore, u_2 can be folded, and again this does not create an edge in $N(u_3)$, so that finally u_3 can be folded. This yields three nodes of degree four (including parallel edges). Applying the reduction rules now either retains at least one node of degree four, or removes at least two further edges. Therefore, we have $\Delta_\varphi(v) \geq 3\lambda - 1\mu$ and $R(G \setminus \{v\})$ contains a node of degree at least four, or $\Delta_\varphi(v) \geq 5\lambda - 1\mu$. Note that neither $G - \{v\}$ nor $G - N[v]$ does contain a tree, since all remaining nodes have a minimum degree of two. □

Theorem 1. Independent Set *can be solved in by Algorithm 1 in time bounded by $O^*(1.1928^{m-n})$. This is $O^*(1.0922^n)$ on cubic graphs.*

Proof. The result is a direct consequence of Lemmata 6–11 for $\lambda = \mu = 1$. Whenever a branching vector is not sufficiently good on its own, a vertex of degree at least four is known to exist, which then allows for a better combined branching:

If G contains a node v of degree at least four, then by Lemmata 6, 7, 8, and 9, branching on v yields a branching vector at least as good as $(4\lambda - \mu, 12\lambda - 5\mu)$ $(5\lambda - \mu, 13\lambda - 7\mu)$, $(5\lambda - \mu, 12\lambda - 5\mu)$, or $(7\lambda - 2\mu, 9\lambda - 4\mu)$.

If G is cubic, by Lemmata 10 and 11 there is $v \in V$ such that branching on v yields a branching vector at least as good as $(5\lambda - \mu, 9\lambda - 4\mu)$, or there is $v \in V$, such that branching on v yields a branching vector at least as good as

$(3\lambda-\mu, 9\lambda-4\mu)$, but $R(G \setminus \{v\})$ contains a node u of degree at least four. In the latter case, we can then branch on u. This either yields a combined branching vector as least as good as $(8\lambda-2\mu, 15\lambda-6\mu, 9\lambda-4\mu)$, $(10\lambda-3\mu, 12\lambda-5\mu, 9\lambda-4\mu)$, or $(7\lambda-2\mu, 16\lambda-6\mu, 9\lambda-4\mu)$, or this yields a combined branching vector at least as good as $(8\lambda-2\mu, 16\lambda-8\mu, 9\lambda-4\mu)$ or $(7\lambda-2\mu, 15\lambda-6\mu, 9\lambda-4\mu)$, but again the first branch yields a graph that contains a node of degree at least four. If we repeat this one more time, we obtain branching vectors that are better than $(7\lambda-2\mu, 16\lambda-6\mu, 9\lambda-4\mu)$, which is good enough.

Optimizing λ and μ as $\lambda = \mu = 1$ yields a run time bound of $O^*(1.192767^{\varphi(G)})$ for Algorithm 1. A reduced cubic graph is three regular, and hence $m = 1.5n$ and $1.192767^{0.5n} \leq 1.092139^n$. □

4 Conclusion

The notion of *satellites* for Maximum Independent Set is a new tool in the toolbox and allows—alongside previously known tools such as *mirrors*, *folding*, or the small separator rule—to tackle many of the harder cases in branching algorithms. Using satellites, we are able to improve the previously best bounds for MIS to $O^*(1.1928^{m-n})$. This is $O^*(1.0922^n)$ on cubic graphs. We are confident that satellites can help to improve the upper bounds for arbitrary graphs as well, but this is still subject to further research.

References

1. P. Alimonti and V. Kann. Some APX-completeness results for cubic graphs. *Theor. Comput. Sci.*, 237(1-2):123–134, 2000.
2. R. Beigel. Finding maximum independent sets in sparse and general graphs. In *Proc. of 10th SODA*, pages 856–857, 1999.
3. P. Berman and T. Fujito. On the approximation properties of independent set problem in degree 3 graphs. In *Proc. of 4th WADS*, volume 955 of *LNCS*, pages 449–460. Springer, 1995.
4. P. Berman and M. Fürer. Approximating maximum independent set in bounded degree graphs. In *Proc. of 5th SODA*, pages 365–371, 1994.
5. N. Bourgeois, B. Escoffier, and V. T. Paschos. An $O^*(1.0977n)$ exact algorithm for max independent set in sparse graphs. In *Proc. of 3rd IWPEC*, number 5018 in LNCS, pages 55–65. Springer, 2008.
6. J. Chen, I. A. Kanj, and G. Xia. Labeled search trees and amortized analysis: Improved upper bounds for np-hard problems. In *Proc. of 14th ISAAC*, number 2906 in LNCS, pages 148–157. Springer, 2003.
7. F. Fomin, F. Grandoni, and D. Kratsch. Measure and conquer: A simple $O(2^{0.288n})$ independent set algorithm. In *Proc. of 17th SODA*, pages 18–25, 2006.
8. F. V. Fomin, F. Grandoni, and D. Kratsch. Measure and conquer: Domination – A case study. In *Proc. of 32nd ICALP*, LNCS, pages 191–203. Springer, 2005.
9. A. M. Frieze and S. Suen. On the independence number of random cubic graphs. *Random Struct. Algorithms*, 5(5):649–664, 1994.
10. M. Fürer. A faster algorithm for finding maximum independent sets in sparse graphs. In *Proc. of 7th LATIN*, number 3887 in LNCS, pages 491–501. Springer, 2006.
11. M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. Freeman, San Francisco, 1979.
12. I. Razgon. A faster solving of the maximum independent set problem for graphs with maximal degree 3. In *Proc. of 2nd ACID*, pages 131–142, 2006.
13. J. M. Robson. Finding a maximum independent set in time $O(2^{n/4})$. Technical Report 1251-01, Université Bordeaux I, LaBRI, 2001.

Aachener Informatik-Berichte

This list contains all technical reports published during the past five years. A complete list of reports dating back to 1987 is available from http://aib.informatik.rwth-aachen.de/. To obtain copies consult the above URL or send your request to: Informatik-Bibliothek, RWTH Aachen, Ahornstr. 55, 52056 Aachen, Email: `biblio@informatik.rwth-aachen.de`

2004-01 * Fachgruppe Informatik: Jahresbericht 2003
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 2004
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
2005-04 Daniel Mölle, Stefan Richter, Peter Rossmanith: A Faster Algorithm for the Steiner Tree Problem
2005-05 Fabien Pouget, Thorsten Holz: A Pointillist Approach for Comparing Honeypots
2005-06 Simon Fischer, Berthold Vöcking: Adaptive Routing with Stale Information
2005-07 Felix C. Freiling, Thorsten Holz, Georg Wicherski: Botnet Tracking: Exploring a Root-Cause Methodology to Prevent Distributed Denial-of-Service Attacks
2005-08 Joachim Kneis, Peter Rossmanith: A New Satisfiability Algorithm With Applications To Max-Cut
2005-09 Klaus Kursawe, Felix C. Freiling: Byzantine Fault Tolerance on General Hybrid Adversary Structures
2005-10 Benedikt Bollig: Automata and Logics for Message Sequence Charts
2005-11 Simon Fischer, Berthold Vöcking: A Counterexample to the Fully Mixed Nash Equilibrium Conjecture

2005-12 Neeraj Mittal, Felix Freiling, S. Venkatesan, Lucia Draque Penso: Efficient Reductions for Wait-Free Termination Detection in Faulty Distributed Systems

2005-13 Carole Delporte-Gallet, Hugues Fauconnier, Felix C. Freiling: Revisiting Failure Detection and Consensus in Omission Failure Environments

2005-14 Felix C. Freiling, Sukumar Ghosh: Code Stabilization

2005-15 Uwe Naumann: The Complexity of Derivative Computation

2005-16 Uwe Naumann: Syntax-Directed Derivative Code (Part I: Tangent-Linear Code)

2005-17 Uwe Naumann: Syntax-directed Derivative Code (Part II: Intraprocedural Adjoint Code)

2005-18 Thomas von der Maßen, Klaus Müller, John MacGregor, Eva Geisberger, Jörg Dörr, Frank Houdek, Harbhajan Singh, Holger Wußmann, Hans-Veit Bacher, Barbara Paech: Einsatz von Features im Software-Entwicklungsprozess - Abschlußbericht des GI-Arbeitskreises "Features"

2005-19 Uwe Naumann, Andre Vehreschild: Tangent-Linear Code by Augmented LL-Parsers

2005-20 Felix C. Freiling, Martin Mink: Bericht über den Workshop zur Ausbildung im Bereich IT-Sicherheit Hochschulausbildung, berufliche Weiterbildung, Zertifizierung von Ausbildungsangeboten am 11. und 12. August 2005 in Köln organisiert von RWTH Aachen in Kooperation mit BITKOM, BSI, DLR und Gesellschaft fuer Informatik (GI) e.V.

2005-21 Thomas Noll, Stefan Rieger: Optimization of Straight-Line Code Revisited

2005-22 Felix Freiling, Maurice Herlihy, Lucia Draque Penso: Optimal Randomized Fair Exchange with Secret Shared Coins

2005-23 Heiner Ackermann, Alantha Newman, Heiko Röglin, Berthold Vöcking: Decision Making Based on Approximate and Smoothed Pareto Curves

2005-24 Alexander Becher, Zinaida Benenson, Maximillian Dornseif: Tampering with Motes: Real-World Physical Attacks on Wireless Sensor Networks

2006-01 * Fachgruppe Informatik: Jahresbericht 2005

2006-02 Michael Weber: Parallel Algorithms for Verification of Large Systems

2006-03 Michael Maier, Uwe Naumann: Intraprocedural Adjoint Code Generated by the Differentiation-Enabled NAGWare Fortran Compiler

2006-04 Ebadollah Varnik, Uwe Naumann, Andrew Lyons: Toward Low Static Memory Jacobian Accumulation

2006-05 Uwe Naumann, Jean Utke, Patrick Heimbach, Chris Hill, Derya Ozyurt, Carl Wunsch, Mike Fagan, Nathan Tallent, Michelle Strout: Adjoint Code by Source Transformation with OpenAD/F

2006-06 Joachim Kneis, Daniel Mölle, Stefan Richter, Peter Rossmanith: Divide-and-Color

2006-07 Thomas Colcombet, Christof Löding: Transforming structures by set interpretations

2006-08 Uwe Naumann, Yuxiao Hu: Optimal Vertex Elimination in Single-Expression-Use Graphs

2006-09 Tingting Han, Joost-Pieter Katoen: Counterexamples in Probabilistic Model Checking

2006-10 Mesut Günes, Alexander Zimmermann, Martin Wenig, Jan Ritzerfeld, Ulrich Meis: From Simulations to Testbeds - Architecture of the Hybrid MCG-Mesh Testbed

2006-11 Bastian Schlich, Michael Rohrbach, Michael Weber, Stefan Kowalewski: Model Checking Software for Microcontrollers

2006-12 Benedikt Bollig, Joost-Pieter Katoen, Carsten Kern, Martin Leucker: Replaying Play in and Play out: Synthesis of Design Models from Scenarios by Learning

2006-13 Wong Karianto, Christof Löding: Unranked Tree Automata with Sibling Equalities and Disequalities

2006-14 Danilo Beuche, Andreas Birk, Heinrich Dreier, Andreas Fleischmann, Heidi Galle, Gerald Heller, Dirk Janzen, Isabel John, Ramin Tavakoli Kolagari, Thomas von der Maßen, Andreas Wolfram: Report of the GI Work Group "Requirements Management Tools for Product Line Engineering"

2006-15 Sebastian Ullrich, Jakob T. Valvoda, Torsten Kuhlen: Utilizing optical sensors from mice for new input devices

2006-16 Rafael Ballagas, Jan Borchers: Selexels: a Conceptual Framework for Pointing Devices with Low Expressiveness

2006-17 Eric Lee, Henning Kiel, Jan Borchers: Scrolling Through Time: Improving Interfaces for Searching and Navigating Continuous Audio Timelines

2007-01 * Fachgruppe Informatik: Jahresbericht 2006

2007-02 Carsten Fuhs, Jürgen Giesl, Aart Middeldorp, Peter Schneider-Kamp, René Thiemann, and Harald Zankl: SAT Solving for Termination Analysis with Polynomial Interpretations

2007-03 Jürgen Giesl, René Thiemann, Stephan Swiderski, and Peter Schneider-Kamp: Proving Termination by Bounded Increase

2007-04 Jan Buchholz, Eric Lee, Jonathan Klein, and Jan Borchers: coJIVE: A System to Support Collaborative Jazz Improvisation

2007-05 Uwe Naumann: On Optimal DAG Reversal

2007-06 Joost-Pieter Katoen, Thomas Noll, and Stefan Rieger: Verifying Concurrent List-Manipulating Programs by LTL Model Checking

2007-07 Alexander Nyßen, Horst Lichter: MeDUSA - MethoD for UML2-based Design of Embedded Software Applications

2007-08 Falk Salewski and Stefan Kowalewski: Achieving Highly Reliable Embedded Software: An empirical evaluation of different approaches

2007-09 Tina Kraußer, Heiko Mantel, and Henning Sudbrock: A Probabilistic Justification of the Combining Calculus under the Uniform Scheduler Assumption

2007-10 Martin Neuhäußer, Joost-Pieter Katoen: Bisimulation and Logical Preservation for Continuous-Time Markov Decision Processes

2007-11 Klaus Wehrle (editor): 6. Fachgespräch Sensornetzwerke

2007-12 Uwe Naumann: An L-Attributed Grammar for Adjoint Code

2007-13 Uwe Naumann, Michael Maier, Jan Riehme, and Bruce Christianson: Second-Order Adjoints by Source Code Manipulation of Numerical Programs

2007-14 Jean Utke, Uwe Naumann, Mike Fagan, Nathan Tallent, Michelle Strout, Patrick Heimbach, Chris Hill, and Carl Wunsch: OpenAD/F: A Modular, Open-Source Tool for Automatic Differentiation of Fortran Codes

2007-15 Volker Stolz: Temporal assertions for sequential and concurrent programs

2007-16 Sadeq Ali Makram, Mesut Güneç, Martin Wenig, Alexander Zimmermann: Adaptive Channel Assignment to Support QoS and Load Balancing for Wireless Mesh Networks

2007-17 René Thiemann: The DP Framework for Proving Termination of Term Rewriting

2007-18 Uwe Naumann: Call Tree Reversal is NP-Complete

2007-19 Jan Riehme, Andrea Walther, Jörg Stiller, Uwe Naumann: Adjoints for Time-Dependent Optimal Control

2007-20 Joost-Pieter Katoen, Daniel Klink, Martin Leucker, and Verena Wolf: Three-Valued Abstraction for Probabilistic Systems

2007-21 Tingting Han, Joost-Pieter Katoen, and Alexandru Mereacre: Compositional Modeling and Minimization of Time-Inhomogeneous Markov Chains

2007-22 Heiner Ackermann, Paul W. Goldberg, Vahab S. Mirrokni, Heiko Röglin, and Berthold Vöcking: Uncoordinated Two-Sided Markets

2008-01 * Fachgruppe Informatik: Jahresbericht 2007

2008-02 Henrik Bohnenkamp, Marielle Stoelinga: Quantitative Testing

2008-03 Carsten Fuhs, Jürgen Giesl, Aart Middeldorp, Peter Schneider-Kamp, René Thiemann, Harald Zankl: Maximal Termination

2008-04 Uwe Naumann, Jan Riehme: Sensitivity Analysis in Sisyphe with the AD-Enabled NAGWare Fortran Compiler

2008-05 Frank G. Radmacher: An Automata Theoretic Approach to the Theory of Rational Tree Relations

2008-06 Uwe Naumann, Laurent Hascoet, Chris Hill, Paul Hovland, Jan Riehme, Jean Utke: A Framework for Proving Correctness of Adjoint Message Passing Programs

2008-07 Alexander Nyßen, Horst Lichter: The MeDUSA Reference Manual, Second Edition

2008-08 George B. Mertzios, Stavros D. Nikolopoulos: The λ-cluster Problem on Parameterized Interval Graphs

2008-09 George B. Mertzios, Walter Unger: An optimal algorithm for the k-fixed-endpoint path cover on proper interval graphs

2008-10 George B. Mertzios, Walter Unger: Preemptive Scheduling of Equal-Length Jobs in Polynomial Time

2008-11 George B. Mertzios: Fast Convergence of Routing Games with Splittable Flows

2008-12 Joost-Pieter Katoen, Daniel Klink, Martin Leucker, Verena Wolf: Abstraction for stochastic systems by Erlang's method of stages

2008-13 Beatriz Alarcón, Fabian Emmes, Carsten Fuhs, Jürgen Giesl, Raúl Gutiérrez, Salvador Lucas, Peter Schneider-Kamp, René Thiemann: Improving Context-Sensitive Dependency Pairs

2008-14 Bastian Schlich: Model Checking of Software for Microcontrollers

2008-15 Joachim Kneis, Alexander Langer, Peter Rossmanith: A New Algorithm for Finding Trees with Many Leaves

2008-16 Hendrik vom Lehn, Elias Weingärtner and Klaus Wehrle: Comparing recent network simulators: A performance evaluation study

2008-17 Peter Schneider-Kamp: Static Termination Analysis for Prolog using Term Rewriting and SAT Solving

2008-18 Falk Salewski: Empirical Evaluations of Safety-Critical Embedded Systems

2009-03 Alexander Nyßen: Model-Based Construction of Embedded Real-Time Software - A Methodology for Small Devices

2009-05 George B. Mertzios, Ignasi Sau, Shmuel Zaks: A New Intersection Model and Improved Algorithms for Tolerance Graphs

2009-07 Joachim Kneis, Alexander Langer, Peter Rossmanith: Derandomizing Non-uniform Color-Coding I

* These reports are only available as a printed version.

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