Ann Oper Res (2006) 146:105–117
DOI 10.1007/s10479-006-0045-4

Experiments on data reduction for optimal domination in networks*

Jochen Alber · Nadja Betzler · Rolf Niedermeier

Published online: 21 June 2006

Abstract We present empirical results on computing optimal dominating sets in networks by means of data reduction through efficient preprocessing rules. Thus, we demonstrate the usefulness of so far only theoretically considered data reduction techniques for practically solving one of the most important network problems in combinatorial optimization.

Keywords Experimental study · Domination · NP-complete problem · Preprocessing by data reduction rules · Optimal solutions · Network optimization

1. Introduction

Domination in networks is one of the most important problems in combinatorial optimization. The underlying NP-complete decision problem DOMINATING SET is defined as follows (Garey and Johnson, 1979):

*Supported by the Deutsche Forschungsgemeinschaft (DFG), project PEAL (parameterized complexity and exact algorithms), NI 369/1 and junior research group PIAF (fixed-parameter algorithms), NI 369/4. Main work done while all authors were with Universität Tübingen. An extended abstract of this paper appeared in the Proceedings of the International Network Optimization Conference (INOC 2003), pages 1–6, held in Evry/Paris, France, October 27–29, 2003.

J. Alber
Power System Applications & Consulting, DIgSILENT GmbH,
Heinrich-Hertz-Str. 9, D-72810 Gomaringen, Germany
e-mail: j.alber@digsilent.de

N. Betzler
Wilhelm-Schickard-Institut für Informatik, Universität Tübingen,
Sand 13, D-72076 Tübingen, Germany
e-mail: betzler@informatik.uni-tuebingen.de

R. Niedermeier (✉)
Institut für Informatik, Friedrich-Schiller-Universität Jena,
Ernst-Abbe-Platz 2, D-07743 Jena, Germany
e-mail: niedermr@minet.uni-jena.de

Input: An undirected graph (network) $G = (V, E)$ and a positive integer k.
Question: Does G have a dominating set of size at most k, i.e., a subset $V' \subseteq V$ of vertices such that every vertex in $V \setminus V'$ is adjacent to some vertex in V'?

The corresponding optimization problem is to determine a dominating set of minimum size. According to a 1998 survey (Kratsch, 1998), there were already more than 200 research papers published on the algorithmic complexity of domination and related network parameters and this number continues to grow. A two-volume book has been published on domination in graphs (Haynes, Hedetniemi, and Slater, 1998a, b). The interest in domination ranges from more fundamental research (e.g., Alber et al., 2002; Fomin and Thilikos, 2003; Haynes et al., 2002) to more applied work (e.g., Sanchis, 2002; Wan, Alzoubi, and Frieder, 2003; Weihe, 1998). In many of the applications, variants of the above given problem are studied. The basic application scenario for domination problems comes from facility location tasks. Intuitively, one might think of the vertices of a minimum dominating set as the most central or most important points of a given network. Besides communication and related networks, other applications arise from voting situations and biological and social network analysis (Roberts, 1979; Valente et al., 2003).

In this piece of work, we empirically investigate the power of data reduction towards *optimally*[1] solving the domination problem on various types of networks. To this end, we take a closer look at and extend a recently introduced theoretical framework of data reduction rules (Alber, Fellows, and Niedermeier, 2004). We implemented and further enriched these rules and we applied them to several random network topologies and experimental data from the literature and various web sites (Chen et al., 2002; Jin, Chen, and Jamin, 2001; Medina et al., 2001; Sanchis, 2002). We show that our data reduction framework in many cases leads efficiently to optimal solutions for realistic networks with up to ten thousands of vertices and edges. As a general rule of thumb, one might say that our data reduction rules usually perform well on sparse networks such as those modeling Internet connectivity. Mostly, the original input instances were significantly reduced to small "hard problem kernels." These remaining networks, usually greatly reduced in size, then can be the starting point for any other algorithmic approach, ranging from exact over approximation to heuristic algorithms. Moreover, we show how our data reduction rules can also be adapted in order to work for directed networks.

2. Algorithmic approach: Data reduction rules

In what follows, we describe various polynomial-time data reduction rules for the DOMINATING SET problem. The idea is to apply the reduction rules over and over again until no further rule will apply. These reduction rules have in common that they explore the local structure of a given network. Depending on this structure, we decide whether a rule is applicable and, if so, the application of a reduction rule may have the following two effects:

1. Determine vertices that can be chosen for an optimal dominating set.
2. Reduce the network by removing edges or vertices.

[1] We mention in passing that DOMINATING SET is hard to approximate. The best known approximation factor achievable by a polynomial-time algorithm is $\Theta(\log n)$ (Feige, 1998). Moreover, observe that in fact our reduction rules to be presented are suitable for solving the optimization problem, not only the decision version as stated above.

It is important to note that whenever vertices or edges are removed from the current instance, this will not affect the size of a minimum dominating set. Additionally, if we decide to choose a vertex to belong to the optimal dominating set we seek for, we may as well remove this vertex from the network and mark all neighbors as being already dominated. Hence, we are left with an instance in which some vertices are already dominated (but still are possible candidates for domination). This brings us to the following generalized problem ANNOTATED DOMINATING SET.

Input: A black-and-white network $G = (B \cup W, E)$, i.e., a network with a set of black vertices B and white vertices W, and a positive integer k.
Question: Is there a $V' \subseteq B \cup W$ with $|V'| \leq k$ such that all *black* vertices are dominated?

We can use this more general model to express an instance in which some vertices (more precisely: the white vertices) are assumed to be already dominated. Initially, the input instance of DOMINATING SET delivers all vertices set black.

2.1. Basic data reduction rules

We revisit two basic reduction rules that were first used in Alber, Fellows, and Niedermeier (2004) in order to show that DOMINATING SET restricted to planar networks admits a so-called linear problem kernel.[2]

The presentation in Alber, Fellows, and Niedermeier (2004), however, purely focuses on the theoretical aspect of problem kernel reduction. Here, in contrast, we will adapt the reduction rules in order to make them applicable for practical purposes. In particular, we will reformulate the rules such that we can deal with the more general ANNOTATED DOMINATING SET problem. The correctness of the following reduction rules is not hard to prove (see Alber, Fellows, and Niedermeier, 2004).

Neighborhood of a single vertex. Consider a vertex $v \in B \cup W$ of the given black-and-white network $G = (B \cup W, E)$. We partition the vertices of the open neighborhood $N(v) := \{ u \in B \cup W \mid \{u, v\} \in E \}$ of v into three different sets:

- the "exit vertices" $N_{\text{exit}}(v)$, through which we can "leave" the closed neighborhood $N[v] := N(v) \cup \{v\}$,
- the "guard vertices" $N_{\text{guard}}(v)$, which are neighbors of exit-vertices, and
- the "prisoner vertices" $N_{\text{prison}}(v)$, which are not direct neighbors of an exit vertex.

More formally, using $N[v] := N(v) \cup \{v\}$, we define

$$N_{\text{exit}}(v) := \{ u \in N(v) \mid N(u) \setminus N[v] \neq \emptyset \},^3$$

$$N_{\text{guard}}(v) := \{ u \in N(v) \setminus N_{\text{exit}}(v) \mid N(u) \cap N_{\text{exit}}(v) \neq \emptyset \},$$

$$N_{\text{prison}}(v) := N(v) \setminus (N_{\text{exit}}(v) \cup N_{\text{guard}}(v)).$$

[2] E.g., in Alber, Fellows, and Niedermeier (2004) it is shown with the help of these reduction rules that every planar network G can be transformed in polynomial time into an instance G', such that G has a dominating set of size k if and only if G' has a dominating set of size k and the size of G' is upper-bounded by $c \cdot ds(G')$, where c is some constant, and $ds(G')$ is the size of an optimal dominating set in G'. Further enhancements can be found in Chen et al. (2005).

[3] For two sets X, Y, where Y is not necessarily a subset of X, we use the convention that $X \setminus Y := \{x \in X : x \notin Y\}$.

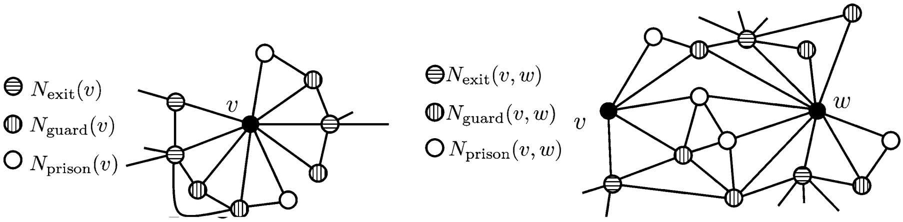

Fig. 1 The left-hand side shows the partitioning of the neighborhood of a single vertex v into the sets $N_{\text{exit}}(v)$, $N_{\text{guard}}(v)$, $N_{\text{prison}}(v)$. Note that the "coloring" in this figure does *not* refer to the colors black and white of the given network. The right-hand side shows the partitioning of the common neighborhood of a pair of vertices v, w into the sets $N_{\text{exit}}(v, w)$, $N_{\text{guard}}(v, w)$, $N_{\text{person}}(v)$

An example which illustrates the partitioning of $N(v)$ into the subsets $N_{\text{exit}}(v)$, $N_{\text{guard}}(v)$, and $N_{\text{prison}}(v)$ can be seen in the left-hand diagram of Fig. 1.

It is clear that a *black* vertex in $N_{\text{prison}}(v)$ can only be dominated by vertices from $\{v\} \cup N_{\text{guard}}(v) \cup N_{\text{prison}}(v)$. Since v will dominate at least as many vertices as any other vertex from $N_{\text{guard}}(v) \cup N_{\text{prison}}(v)$, it is safe to place v into the optimal dominating set we seek for.

Main Rule 1. *If $N_{\text{prison}}(v) \cap B \neq \emptyset$ for $v \in B \cup W$, then it is optimal to choose v to belong to the dominating set:*

- *remove v from G and color all neighbors of v white, and*
- *remove $N_{\text{guard}}(v)$ and $N_{\text{prison}}(v)$ from G.*

Neighborhood of a pair of vertices. Similar to Rule 1, we explore the union of the neighborhoods $N(v, w) := N(v) \cup N(w)$ of two vertices $v, w \in V$. Analogously, we now partition $N(v, w)$ into three disjoint subsets $N_{\text{exit}}(v, w)$, $N_{\text{guard}}(v, w)$, and $N_{\text{prison}}(v, w)$. Setting $N[v, w] := N[v] \cup N[w]$, we define

$$N_{\text{exit}}(v, w) := \{ u \in N(v, w) \mid N(u) \setminus N[v, w] \neq \emptyset \},$$

$$N_{\text{guard}}(v, w) := \{ u \in N(v, w) \setminus N_{\text{exit}}(v, w) \mid N(u) \cap N_{\text{exit}}(v, w) \neq \emptyset \},$$

$$N_{\text{prison}}(v, w) := N(v, w) \setminus (N_{\text{exit}}(v, w) \cup N_{\text{guard}}(v, w)).$$

The right-hand diagram of Fig. 1 shows an example which illustrates the partitioning of $N(v, w)$ into the subsets $N_{exit}(v, w)$, $N_{guard}(v, w)$, and $N_{prison}(v, w)$.

Our second reduction rule—compared to Rule 1—is slightly more complicated, but it is based on the same principle: We try to detect an optimal domination of the black prisoner vertices $N_{prison}(v, w) \cap B$ in our local structure $N(v, w)$. It is clear that a *black* vertex in $N_{prison}(v, w)$ can only be dominated by vertices from $\{v, w\} \cup N_{guard}(v, w) \cup N_{prison}(v, w)$. The following rule determines cases in which it is safe to choose one of the vertices v or w (or both) to belong to the optimal dominating set we seek for. The correctness of this reduction rule is not hard to prove (see Alber, Fellows, and Niedermeier, 2004).

Main Rule 2. *Consider $v, w \in V$ ($v \neq w$) and suppose that $N_{\text{prison}}(v, w) \cap B \neq \emptyset$. Suppose that $N_{\text{prison}}(v, w) \cap B$ cannot be dominated by a single vertex from $N_{\text{guard}}(v, w) \cup N_{\text{prison}}(v, w)$.*

Case 1. *If $N_{prison}(v, w) \cap B$ can be dominated by a single vertex from $\{v, w\}$:*

(1.1) *If $N_{prison}(v, w) \cap B \subseteq N(v)$ as well as $N_{prison}(v, w) \cap B \subseteq N(w)$, then it is optimal to choose v or w (or both), but the decision for one of these choices cannot yet be made, hence:*

- *as a gadget we add three new black vertices z, z', z'' and six new edges $\{v, z\}$, $\{w, z\}$, $\{v, z'\}$, $\{w, z'\}$, $\{v, z''\}$, $\{w, z''\}$ to G and*
- *remove $N_{prison}(v, w)$ and $N_{guard}(v, w) \cap N(v) \cap N(w)$ from G.*

(1.2) *If $N_{prison}(v, w) \cap B \subseteq N(v)$, but not $N_{prison}(v, w) \cap B \subseteq N(w)$, then it is optimal to choose v:*

- *remove v from G and color all neighbors of v white and*
- *remove $N_{prison}(v, w)$ and $N_{guard}(v, w) \cap N(v)$ from G.*

(1.3) *If $N_{prison}(v, w) \cap B \subseteq N(w)$, but not $N_{prison}(v, w) \subseteq N(v)$, then it is optimal to choose w: proceed as in* (1.2) *with roles of v and w interchanged.*

Case 2. *If $N_{prison}(v, w)$ cannot be dominated by a single vertex from $\{v, w\}$, then it is optimal to chose both v and w:*

- *remove v and w from G and color all their neighbors white and*
- *remove $N_{prison}(v, w)$ and $N_{guard}(v, w)$ from G.*

It is not hard to see that Reduction Rules 1 and 2 lead to an optimal dominating set and they can be carried out in polynomial time; more precisely, it can be shown that the application of Rule 1 and 2 takes $O(|V|^3)$ and $O(|V|^4)$ time in the worst case, respectively (see Alber, Fellows, and Niedermeier, 2004).[4] Our basic reduction then processes the graph by choosing all possible (pairs of) vertices until no rule is applicable any more. Observe that for efficiency reasons, one prefers to apply Rule 1 as long as possible and afterwards continues with Rule 2. It may happen, then, that after Rule 2 again Rule 1 applies due to the new graph structure caused by Rule 2.

2.2. Further data reduction rules

The original versions of the above two data reduction rules turned out to be sufficient for theoretical purposes, i.e., they were sufficient for proving a linear problem kernel on planar networks (Alber, Fellows, and Niedermeier, 2004). Note that, so far, a network which consists of white vertices only will not be reduced any further. The following rules, which were basically introduced in Alber et al. (2005) as a tool in the theoretical analysis of a search tree algorithm for DOMINATING SET on planar networks, will also cover some further, easy cases. Notably, they lead to significant further improvements in our experimental analysis to follow.

[4]We remark that these running times are pure worst-case estimates and turn out to be much better on average in our experimental studies. In particular, for practical purposes it is important to see that Rule 2 can only be applied for vertex pairs that are at distance at most three. Also, using elaborate algorithmic techniques an improvement of the worst-case complexity of the rules seems possible.

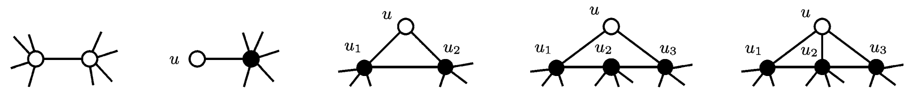

Fig. 2 Illustration of the settings in which Simple Rules 1, 2, 3, and 4 apply. Grey vertices may be either black or white

Simple Rule 1. Delete edges between white vertices.
Simple Rule 2. Let u be a white vertex of degree at most 1. Then, delete u.
Simple Rule 3. Let u be a white vertex of degree 2, with two black neighbors u_1 and u_2.

(3.1) If u_1 and u_2 are connected by an edge, then delete u.
(3.2) If u_1 and u_2 are connected via a third (black or white) vertex u_3, then delete u.

Simple Rule 4. Let u be a white vertex of degree 3, with three black neighbors u_1, u_2, and u_3. If the edges $\{u_1, u_2\}$ and $\{u_2, u_3\}$ are present in G (and possibly also $\{u_1, u_3\}$), then delete u.

Figure 2 illustrates the various settings where these rules apply. It is not hard to verify that all four reduction rules are correct.

2.3. Dealing with Directed Dominating Set

In several applications we have to deal with directed networks. Here, a vertex v is dominated iff it is in the dominating set or if there is an arc (u, v) (i.e., v is an outgoing neighbor of u) and u is in the dominating set. In order to cope with such settings, we here describe a transformation from Directed Dominating Set to (undirected) Annotated Dominating Set.

Let $G = (V, A)$ be an instance of Directed Dominating Set. Then we construct an undirected black-and-white network $G' = (B \cup W, E)$ as follows:

$$B := \{u' \mid u \in V\}, \quad W := \{u'' \mid u \in V\}, \quad \text{and}$$
$$E := \{\{u', u''\} \mid u \in V\} \cup \{\{u'', v'\}, \{u'', v''\} \mid (u, v) \in A\}.$$

In other words, every vertex u in G is duplicated with a black copy u' (which enforces that u needs to be dominated) and a white copy u'' (which simulates the choice of u to belong to a dominating set). We add (undirected) edges connecting u'' with u' and with all outgoing neighbors of u in the directed network.

It is easy to see that G admits an optimal directed dominating set of size k if and only if G' admits an optimal annotated dominating set of size k: Suppose that D is an optimal dominating set in G, then $D'' := \{u'' \mid u \in V\}$ is an optimal dominating set in G'. Conversely, suppose that D' is an optimal dominating set in G'. Since D' is assumed to be optimal and since in G' we have $N[u'] \subseteq N[u'']$, we may assume that $D' \cap \{u' \mid u \in V\} = \emptyset$ for all $u \in V$ (otherwise we might interchange u' with u''). But then D' induces a directed dominating set $D := \{u \mid u' \in D'\}$ for G.

Clearly, in order to find an optimal dominating set for a directed graph G, we can use the above transformation[5] and then apply our undirected reduction rules to the transformed instance G'.

3. Experimental results

We tested our data reduction framework on random planar networks and various network data provided in the literature and publically available on the web. Among the publically available networks, we firstly investigate networks obtained from (Internet) topology generators (Inet, BRITE) (Jin, Chen, and Jamin, 2001; Medina et al., 2001) and networks of autonomous systems (Chen et al., 2002). Besides many others, one possible interest in computing small dominating sets in Internet networks might be the placement of time servers (for NTP protocol synchronization). Note that a dominating set in the Internet topology denotes a minimum number of locations from which the whole network can be reached quickly by a single link (for instance, to supply each node with the current time signal). Secondly, we turn our attention to a network generated from the field of biochemistry: We consider a protein-protein interaction network generated by using the database BIND (Bader et al., 2001). And thirdly, we will have a look at three examples of directed networks (an HTML network and two food web networks from biology). All our experiments have been run on a 2.4 GHz Linux AMD Athlon 64 3400+ PC with 1 GB main memory. The code has been implemented in C++ using the algorithm library LEDA (Mehlhorn and Näher, 1999).

Random planar networks. We start our experimental analysis with combinatorial random planar networks. These networks have been generated with the standard function provided by the algorithm library LEDA (Mehlhorn and Näher, 1999).[6] More precisely, we created three sample sets of 100 random planar networks each, containing instances with 500, 1500, and 4000 vertices. The main reason for restricting ourselves to planar networks (which might be of less "real" practical interest than the other investigations to follow) is that the original reduction rules were defined and analyzed for planar networks (Alber et al., 2005; Alber, Fellows, and Niedermeier, 2004). Observe that the afterwards considered network instances as provided by Internet topology generators etc. clearly are non-planar. Our main goal here is to give a first impression of the strength of the data reduction rules. We also demonstrate that adding the Simple Rules from Section 2.2 to the Main Rules from Section 2.1 really pays off—from a purely theoretical point of view the Simple Rules are superfluous. In addition, here we study the effects of each single rule and their interplay which—by "cascading effects"—greatly improves the obtained results.

The potential of the aforementioned reduction rules was tested individually. We ran a series of tests using Rule 1 only, using Rule 2 only, using a combination of Rule 1 and Rule 2, and, finally, using Rules 1 and Rule 2 together with the four Simple Rules.

For each test run, we measured the following figures:

- *# vertices removed:* the number of vertices removed by the data reduction;

[5]We remark, however, that the given transformation is not planarity-preserving.

[6]For each instance with n vertices, first a "maximal planar network" with $3n-6$ edges is randomly generated, then a number m between $n-1$ and $3n-6$ is randomly chosen and all but m edges are removed from the network. We remark that this method does not generate graphs according to the uniform distribution (see Mehlhorn and Näher, 1999 for details).

Table 1 Summary of experimental results for combinatorial random planar network instances. The numbers in the various rows are taken as the *average* over networks in PGn of the corresponding column, where PGn stands for combinatorially random planar networks (created with the standard LEDA function) with n vertices

Sample set	PG500	PG1500	PG4000
# total vertices ‡	479.1	1433.4	3840.9
# total edges	1005.1	2875.1	8075.2
Size of optimal ds	99.7	311.5	791.8
Rule 1:			
# vertices removed	336.4	998.0	2722.0
(percentage)	70.2	69.6	70.9
# edges removed	801.3	2281.9	6523.9
(percentage)	79.7	79.3	80.8
# vertices for DS found	77.6	243.7	620.0
(percentage)†	77.9	78.2	78.3
time (sec)	0.052	0.42	2.04
Rule 2:			
# vertices removed	351.8	1019.5	2750.0
(percentage)	73.4	71.1	72.7
# edges removed	880.4	2483.0	7050.0
(percentage)	87.6	86.3	87.3
# vertices for DS found	70.2	212.6	547.8
(percentage)†	70.4	68.3	69.2
time (sec)	0.129	1.07	5.26
Rule 1 + 2:			
# vertices removed	377.8	1107.7	3015.9
(percentage)	78.9	77.3	78.5
# edges removed	892.1	2521.6	7169.1
(percentage)	88.8	87.7	88.8
# vertices for DS found	85.9	266.0	678.7
(percentage)†	86.2	85.4	85.7
time (sec)	0.12	1.83	6.74
Rule 1 + 2 + Simple Rules 1, 2, 3, 4:			
# vertices removed	475.6	1424.6	3814.3
(percentage)	99.2	99.4	99.3
# edges removed	999.7	2863.25	8035.8
(percentage)	99.5	99.6	99.5
# vertices for DS found	98.8	309.1	785.3
(percentage)†	99.1	99.2	99.2
time (sec)	0.05	0.32	1.69

† percentage with respect to an optimal dominating set.
‡ number of vertices without degree 0.

- *# edges removed:* the number of edges removed by the data reduction;
- *# vertices for DS found:* the number of vertices that could be determined to be in an optimal dominating set;
- the time in seconds needed in order to reduce the network with respect to the given set of rules.

The results of the tests are summarized in Table 1. Using the combination of Rules 1 and 2, as they were used to prove the linear problem kernel, we may say that, in average over 100 networks each time,

- more than 77% of the vertices and
- more than 87% of the edges

were removed from the network. Both rules together determined a very high percentage (in average, more than 85%) of the vertices of an optimal dominating set—seemingly independent of the size of the input networks. The overall running time for the reduction ranged from less than one second (for small network instances with 500 vertices) to around 7 seconds (for larger network instances with 4000 vertices). Surprisingly, using Rule 1 *or* Rule 2 *alone* already resulted in a very powerful data reduction. In both cases, in average, more than 70% of the vertices could be removed from the network. Clearly, reducing a network with respect to Rule 1 is less time-consuming than reducing a network with respect to the more complex Rule 2. Moreover, Rule 1 seemed to be stronger than Rule 2 in the detection of vertices of an optimal dominating (in average, 78% compared to 69%). Conversely, we noticed a subtle tendency that Rule 2 removes more edges compared to Rule 1 (in average, 87% compared to 80%).

Finally, enriching Rules 1 and 2 with the four Simple Rules led to an extremely powerful data reduction on our set of random instances. Most interestingly, the combination of these rules removed, on average,

- more than 99.2% of the vertices and
- more than 99.5% of the edges

of the original network. More than 99.1% of the vertices that belong to an optimal dominating set have been detected. These percentages again seem to be independent of the size of the input network. We observed that in this extended setting, in average the running times for the data reduction went down to less than 0.05 seconds (for networks of 500 vertices) and less than two seconds (for networks of 4000 vertices). This is also due to the fact that we applied simple reduction rules first before more complicated rules (such as Rule 2) were applied. Thus, the time-consuming sophisticated steps usually had to be carried out on small networks only.

Autonomous systems networks. Chen et al. (2002) provided network data concerning Internet connectivity at the level of autonomous systems (AS). They report on "AS connectivity maps" obtained from routing tables collected by the Oregon route server on many different dates and they argue why these may provide an incomplete picture of the physical connectivity that exists in the actual Internet. They finally present a network model and refined connectivity maps that are supposed to provide a more complete picture of the Internet connectivity (see Chen et al., 2002 for any details). Thus, one arrives at two sets of network data supposed to model the (time) varying Internet structure, the "Oregon data" and the more refined data proposed in Chen et al. (2002). We took both data sets and applied our data reduction techniques to compute minimum size dominating sets in these networks of more than 10000 vertices and around 20000 (old model) and 30000 (new model) edges. For both cases, we either could already compute an optimal dominating set or, in few cases, were left with a tremendously reduced network where one could easily compute the remaining optimal domination vertices by brute-force methods. Table 2 lists the results for the old model (here the computation per network took about two seconds) and the new model (here the computation per network took between three and four seconds). Interestingly, the sizes of the optimal dominating sets seem to be rather stable slightly below 1000 in all networks (old and new). The new model (with almost 50% more edges) seems to yield only slightly smaller domination numbers.

Table 2 *Autonomous Systems Networks:* Experimental results for the AS networks as obtained from routing tables collected by the Oregon route server at different dates, using both models—the standard model ("Oregon") and the enriched model ("Oregon+") from Chen et al. (2002)

	AS model: Oregon				Enriched AS model: Oregon+			
Date	Vertices	Edges	% reduced	DS	Vertices	Edges	% reduced	DS
03/31/01	10670	22002	100	956	10900	31180	100.00	933
04/07/01	10729	21999	100	968	10981	30855	100.00	933
04/14/01	10790	22469	100	977	11019	31761	>99.99	946
04/21/01	10895	22747	100	981	11080	31538	>99.99	954
04/28/01	10886	22493	100	990	11113	31434	100.00	962
05/05/01	10943	22607	100	988	11157	30943	>99.99	959
05/12/01	11011	22677	100	988	11260	31303	>99.99	959
05/19/01	11051	22724	100	978	11375	32287	>99.99	965
05/26/01	11174	23409	100	992	11461	32730	>99.99	960

The columns show the size of the different networks (i.e., number of vertices and number of edges), and the amount by which our reduction rules reduced the given network. In addition, the last column reports on the size of the minimum dominating set (DS) as computed by our method.

Networks from topology generators. Here we report on results using network data produced by the Internet topology generators Inet (Jin, Chen, and Jamin, 2001) and BRITE (Medina et al., 2001). We refer to the given papers for any details concerning the data generation process.

Table 3 gives our results and the parameter settings we used for generating the corresponding networks from Inet 2.0. We only emphasize few of our experimental findings. It is striking that except for one all networks could be completely resolved for up to 10000 vertices and usually more than twice as many edges.[7] The dominating set sizes were inbetween 801 and 2129. We observed a decrease in the size of the optimal dominating set when considering networks of higher density. Besides, for fixed density, the size of an optimal dominating set behaved almost proportional to the number of vertices in the networks produced by the topology generators.

In contrast to the results for the Inet and AS networks, applied to BRITE networks our data reduction rules in most cases had only little or no effect at all. We created five sample sets using the BRITE generator with different combinations of the settings Node Placement Strategy (NPS), Growth Type (GT), and Preference Connectivity (PC). Each set consists of 5 networks with 1000 vertices and 1997 edges. Only two of all instances could be reduced to a size of less than 100 vertices by applying our reduction rules. In all other cases the number of remaining vertices ranged between 765 and 1000. The running times were even higher than for the Inet networks as the more sophisticated rules had to be applied more often to the full network. Note that one of the instances that was largely reduced was created by settings which are considered to model the Internet in a particularly realistic way.

Protein-protein interaction networks. As a typical example for a biochemical relevant network, we consider the protein-protein interaction network of yeast, which was generated

[7]The running times grow with the number of vertices and the density of the networks from less than half a second up to 2 seconds for the largest network. The exceptionally high running time of more than 8 seconds for the instance with 5000 vertices and density parameter $d = 0.001$ is due to the fact that here the more complicated data reduction rules had to be applied more often than for the other networks.

Table 3 *Inet 2.0 Topology Generator:* The table summarizes the performance of the data reduction on various networks generated with the Inet topology generator (Jin, Chen, and Jamin, 2001). We constructed networks of 5000, 7500, and 10000 vertices using the default configuration and varying over the parameter *d* (expressing the fraction of low-degree vertices, see Jin, Chen, and Jamin (2001) for details) in order to obtain networks with various numbers of edges. The columns show the performance of our data reduction, reporting on the time needed, the amount by which the networks were reduced, and the size of an optimal dominating set (DS) as computed by our method

	Inet: 5000 vertices				Inet: 7500 vertices				Inet: 10000 vertices			
Parameter	Edges	Time	Reduced %	DS	Edges	Time	Reduced %	DS	Edges	Time	Reduced %	DS
d: 0.5	9121	0.35	100	1085	13811	0.56	100	1650	18532	0.74	100	2129
d: 0.3	10434	0.43	100	1062	15765	0.69	100	1584	21145	0.93	100	2101
d: 0.2	11084	0.65	100	993	16758	0.97	100	1483	22451	2.05	100	1955
d: 0.1	11470	1.17	100	900	17733	1.57	100	1356	23764	3.44	100	1802
d: 0.05	12066	1.48	100	847	18225	1.94	100	1265	24416	5.68	100	1699
d: 0.001	12383	8.26	99.9	801	18712	5.22	100	1183	25045	7.00	100	1541

by the data from the database BIND (Bader et al., 2001). The network consists of 4919 vertices corresponding to the proteins and 17152 edges for the protein-protein interactions. In less than three seconds we could determine an optimal dominating set consisting of 860 proteins. Finding a dominating set in a protein-protein interaction network could be useful to infer its functionality. The proteins belonging to a dominating set can be considered as important vertices as they are able to interact with all proteins of the network. Therefore, it is likely that the dominating set comprises proteins that perform important regulatory functions.

Some directed networks (HTML networks and food webs). Finally, to gain first insights for directed networks, we also tested our rules on the proposed translation (see Section 2.3) of directed networks into undirected ones. We describe results for three particular networks. Firstly, we considered an HTML network which contained 739 vertices and 3447 arcs. This network was created by taking the HTML document SELFHTML, Version 7.0 (an HTML tutorial) available from `http://selfaktuell.teamone.de`. Pages have been translated into vertices and links have been translated into arcs. Within 0.15 seconds our reduction rules computed an optimal dominating set of size 137. Thus, this dominating set contains the minimum amount of pages from which each other page of the HTML document can be reached following only one link (i.e., by one click).

Secondly, we considered two food web networks from biology (from `http://www.cosin.org/` network data sets), where an arc points from prey to predator. We considered the Silwood Park food web with 308 vertices and 884 arcs. In less than a second an optimal dominating set of only 24 preys was determined. The second food web we tested is from the Ythan Estuary consisting of 270 vertices and 1286 arcs. In this case, after about 0.3 seconds we obtained 13 preys that are part of an optimal dominating set. We were left with a reduced network of 25 vertices and 51 arcs where no more reduction rule applied. Within few more seconds, using a tree decomposition based algorithm (Alber et al., 2002) we determined the remaining vertices of an optimal dominating set such that the optimal dominating set of the whole food web consisted of 17 preys. Interestingly, both food webs have fairly small domination numbers. Here, for instance, an optimal dominating set can be interpreted as a minimum size set of preys whose disappearance would affect the menu of all predators.

Further investigations on various directed networks (e.g., also on social networks as discussed in Valente et al. (2003)) remain to be conducted in future work.

4. Conclusion and Outlook

In this piece of work, we demonstrated the impressive potential of comparatively simple and easy to efficiently implement data reduction rules in order to compute optimal dominating sets in realistic networks up to sizes of ten thousands of vertices and edges. In many cases, the problem was completely solved, yielding dominating sets of minimum size. Otherwise, usually a significant reduction of the size of the input data was achieved. Our main conclusion is that data reduction should become a tool for everyone dealing with domination in networks. A more comprehensive picture of the whole scenario (also showing irreducible graphs) can be found in Alber (2003). Data reduction rules can be easily be combined also with purely heuristic or approximation approaches towards computing small dominating sets in networks.

On the "negative" side, our data reduction rules seem to behave poorly when applied to dense networks with many edges. Sanchis (2002) generates these sorts of data and proposes heuristic algorithms to compute not necessarily optimal dominating sets in these settings. However, for many "real-world" sparse networks our data reduction scenario performed extremely well.

Since our data reduction rules evolved out of purely theoretical research (Alber et al., 2005; Alber, Fellows, and Niedermeier, 2004), our work also provides an excellent example for a fruitful link between theoretical and practical computer science.

As to future work, we would like to further extend the range of networks which are amenable to the data reduction rules. To this end, one needs to investigate network structures that so far seem to be resistant to the given rules. For instance, whereas most sparse networks seem to be no problem for our data reduction framework, grid-like networks still remain hard where little data reduction seems achievable by the use of the presented rules. In addition, it remains to establish and investigate similar reduction rules for practically relevant variants of DOMINATING SET such as CONNECTED DOMINATING SET (Wan, Alzoubi, and Frieder, 2003; Demaine and Hajiaghayi, 2005) or POWER DOMINATING SET (Haynes et al., 2002; Guo, Niedermeier, and Raible, 2005). Although these problems are computationally hard from a theoretical point of view (concerning exact as well as approximation solutions), data reduction rules seem to indicate a fruitful and theoretically well-founded way towards attacking these important network optimization problems.

Acknowledgments We thank Frederic Dorn for his efforts in implementing the data reduction rules and Britta Dorn for useful discussions on the directed version of dominating set. Besides, we are grateful to the anonymous referees of *INOC 2003* and Falk Hüffner for comments that helped to improve the presentation of the paper.

References

Alber, J. (2003). *Exact Algorithms for NP-hard Problems on Networks: Design, Analysis, and Implementation*. PhD thesis, WSI für Informatik, Universität Tübingen, Germany.

Alber, J., H.L. Bodlaender, H. Fernau, T. Kloks, and R. Niedermeier. (2002)."Fixed Parameter Algorithms for Dominating Set and Related Problems on Planar Graphs." *Algorithmica*, 33(4), 461–493.

Alber, J., H. Fan, M.R. Fellows, H. Fernau, R. Niedermeier, F. Rosamond, and U. Stege. (2005). "A Refined Search Tree Technique for Dominating Set on Planar Graphs." *Journal of Computer and System Sciences*, 71(4):385–405.

Alber, J., M.R. Fellows, and R. Niedermeier. (2004).“Polynomial Time Data Reduction for Dominating Set.” *Journal of the ACM*, 51(3), 363–384.

Bader, G.D., I. Donaldson, C. Wolting, B.F.F. Quellette, T. Pawson, and C.W.V. Hogue. (2001).“BIND—The Biomolecular Interaction Network Database.” *Nucleic Acids Research*, 29(1), 242–245.

Chen, J., H. Fernau, I.A. Kanj, and G. Xia. (2005). “Parametric Duality and Kernelization: Lower Bounds and upper Bounds on Kernel Size.” In *Proc. 22d STACS*, vol. 3404 of *LNCS*, Springer, pp. 269–280.

Chen, Q., H. Chang, R. Govindan, S. Jamin, S. Shenker, and W. Willinger. (2002).“The Origin of Power-Laws in Internet Topologies Revisited.” In *Proc. of INFOCOM*.

Demaine, E.D. and M. Hajiaghayi. (2005).“Bidimensionality: New Connections Between FPT Algorithms and PTASs.” In *Proc. 16th SODA*, ACM/SIAM, pp. 590–601.

Feige, U. (1998).“A Threshold of $\ln n$ for Approximating Set Cover.” *Journal of the ACM*, 45(4):634–652.

Fomin, F.V. and D.M Thilikos. (2003).“Dominating Sets in Planar Graphs: Branch-Width and Exponential Speed-Up.” In *Proc. 14th SODA*, ACM/SIAM, pp. 168–177.

Garey, M.R. and D.S. Johnson. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman.

Guo, J., R. Niedermeier, and D. Raible. (2005).“Improved Algorithms and Complexity Results for Power Domination in Graphs.” In *Proc. 15th FCT*, volume 3623 of *LNCS*, Springer, pp. 172–184.

Haynes, T.W., S.M. Hedetniemi, S.T. Hedetniemi, and M.A. Henning. (2002).“Domination in Graphs Applied to Electric Power Networks.” *SIAM Journal on Discrete Mathematics*, 15(4), 519–529.

Haynes, T.W., S.T. Hedetniemi, and P.J. Slater. (1998a). *Domination in Graphs: Advanced Topics*, vol. 209 of *Pure and Applied Mathematics*. Marcel Dekker.

Haynes, T.W., S.T. Hedetniemi, and P.J. Slater. (1998b). *Fundamentals of Domination in Graphs*, vol. 208 of *Pure and Applied Mathematics*. Marcel Dekker.

Jin, C., Q. Chen, and S. Jamin. (2001).“Inet: Internet Topology Generator.” Technical Report CSE-TR443-00, Department of EECS, University of Michigan.

Kratsch, D. (1998).“Algorithms.” In T.W. Haynes, S.T. Hedetniemi, and P.J. Slater, (eds.), *Domination in Graphs: Advanced Topics*. Marcel Dekker.

Medina, A., A. Lakhina, I. Matta, and J.W. Byers. (2001).“Brite: An Approach to Universal Topology Generation.” In *Proc. MASCOTS*.

Mehlhorn, K. and S. Näher. (1999). *LEDA: A Platform of Combinatorial and Geometric Computing*. Cambridge University Press.

Roberts, F.S. (1979). *Graph Theory and Its Applications to Problems of Society*. SIAM Press, Odyssey Press.

Sanchis, L.A. (2002).“Experimental Analysis of Heuristic Algorithms for the Dominating Set problem.” *Algorithmica*, 33(1), 3–18.

Valente, T.W., B.R. Hoffman, A. Ritt-Olson, K. Lichtman, and C.A. Johnson. (2003).“Strategies on Peer-Led Tobacco Prevention Programs in Schools.” *American Journal of Public Health*, 93(11), 1837–1843.

Wan, P.-J., K.M. Alzoubi, and O. Frieder. (2003).“A Simple Heuristic for Minimum Connected Dominating Set in Graphs.” *International Journal of Foundations of Computer Science*, 14(2), 323–333.

Weihe, K. (1998).“Covering Trains by Stations or the Power of Data Reduction.” In *Proc. 1st ALEX’98*, pp. 1–8.