Proc. Int. Network Optimization Conf. (INOC 2003), pp. 1–6, 2003

Experiments on Data Reduction for Optimal Domination in Networks*

Jochen Alber Nadja Betzler Rolf Niedermeier
Wilhelm-Schickard-Institut für Informatik, Universität Tübingen
Sand 13, D-72076 Tübingen, Fed. Rep. of Germany
{alber,betzler,niedermr}@informatik.uni-tuebingen.de

Abstract

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

keywords Graph Theory, Location, Topological Design, Network Models, Operations Research

1 Introduction

Domination in networks is one of the the most important problems in combinatorial optimization. The underlying NP-complete decision problem Dominating Set is defined as follows:
Input: A 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 $V - V'$ is adjacent to some vertex in V'?
The corresponding optimization problem is to determine a dominating set of minimum size. A two-volume book has been published on domination in graphs [6, 7]. The interest in domination ranges from more fundamental research to more applied work (e.g., [5, 11, 13]). In many of the applications, variants of the 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 [10, 12].

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 reduction rules [2]. We implemented and further enriched these rules and we applied them to several network topologies and experimental data from the literature and from various web sites [3, 8, 9]. Our data reduction framework in many cases leads efficiently to optimal solutions for realistic networks with up to ten thousands of vertices and edges. Moreover, we show how our data reduction rules can be transformed in order to work for directed networks. Altogether, the two main contributions of this work are to experimentally validate the usefulness of existing data reduction rules in a novel combination [1, 2] and to show that the given theoretical framework also generalizes to networks with directed edges.

2 Algorithmic Approach: Reduction Rules

In what follows, we describe various polynomial-time reduction rules for the Dominating Set problem. The idea is to apply the reduction rules over and over (alternatingly) 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: They determine vertices that can be chosen for an optimal dominating set and reduce the network by removing edges or

*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.

[1] We mention in passing that Dominating Set is hard to approximate. The best known approximation factor achievable by a polynomial-time algorithm is $\Theta(\log n)$ [4]. 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.

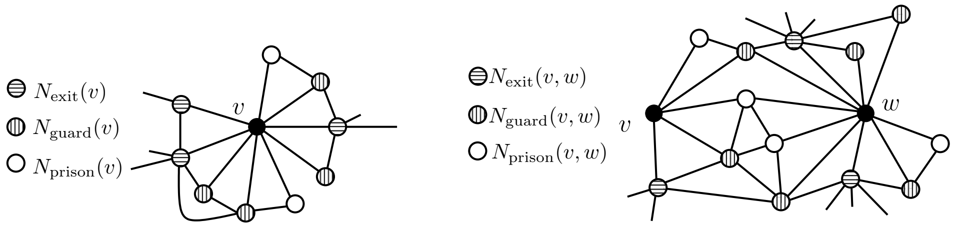

Figure 1: The left-hand side shows the partitioning of the neighborhood of a single vertex v into the sets $N_{\text{exit}}(v), N_{\text{guard}}(v), N_{\text{prison}}(v)$. Note that the "coloring" in this figure does *not* refer to the colors black and white of the given network. The right-hand side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

vertices. It is important to note that by these rules we can only guarantee at least one optimal dominating set but give up to find all of them. In this way, 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 $G = (B \uplus W, E)$ with black vertices B and white vertices W, and a positive integer k.

Question: Is there a $V' \subseteq B \uplus W$ 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.

Basic Reduction Rules. We revisit two basic reduction rules that were first used in [2] where it was shown that Dominating Set restricted to planar networks admits a so-called linear problem kernel. The presentation in [2], however, purely focuses on the theoretical aspect of problem kernel reduction. Here, in contrast, we will slightly modify 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 [2]).

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

$$\begin{aligned} N_{\text{exit}}(v) &:= \{ u \in N(v) \mid N(u) \setminus N[v] \neq \emptyset \}, \\ N_{\text{guard}}(v) &:= \{ u \in N(v) \setminus N_{\text{exit}}(v) \mid N(u) \cap N_{\text{exit}}(v) \neq \emptyset \}, \\ N_{\text{prison}}(v) &:= N(v) \setminus (N_{\text{exit}}(v) \cup N_{\text{guard}}(v)). \end{aligned}$$

An example demonstrating the partitioning of the neighborhood of a single vertex is given in 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 $N_{prison}(v) \cap B \neq \emptyset$ for $v \in B \uplus W$ 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 $N_{guard}(v)$ and $N_{prison}(v)$ 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 $v, w \in B \uplus W$. Analogously, we now partition $N(v,w)$ into three disjoint subsets. Setting $N[v,w] := N[v] \cup N[w]$, we define

$$\begin{aligned} N_{\text{exit}}(v,w) &:= \{ u \in N(v,w) \mid N(u) \setminus N[v,w] \neq \emptyset \}, \\ N_{\text{guard}}(v,w) &:= \{ u \in N(v,w) \setminus N_{\text{exit}}(v,w) \mid N(u) \cap N_{\text{exit}}(v,w) \neq \emptyset \}, \\ N_{\text{prison}}(v,w) &:= N(v,w) \setminus (N_{\text{exit}}(v,w) \cup N_{\text{guard}}(v,w)). \end{aligned}$$

Following the second reduction rule below also consider the example of a partitioning of the joint neighborhood of two vertices as given in Fig. 1.

The idea is to detect an optimal domination of the black prisoner vertices $N_{\text{prison}}(v,w) \cap B$ in our local structure $N(v,w)$. It is clear, that a *black* vertex in $N_{\text{prison}}(v,w)$ can only be dominated by vertices from $\{v,w\} \cup N_{\text{guard}}(v,w) \cup N_{\text{prison}}(v,w)$. 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.

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

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 two new white vertices z, z' and edges $\{v,z\}, \{w,z\}, \{v,z'\}, \{w,z'\}$ 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 $N_{prison}(v,w) \cap B \subseteq N(v)$, 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 choose 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 Main Rules 1 and 2 lead to an optimal dominating set and they can be carried out in time $O(|V|^3)$ and $O(|V|^4)$, respectively (see [2]).[2] Our basic reduction then processes the graph by choosing all possible (pairs of) vertices until no more application of one of the rules is possible. Observe, that for efficiency reasons, one prefers to apply Rule 1 as long as possible and the continue 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.

Further Reduction Rules. The original versions of the above two reduction rules turned out to be sufficient for theoretical purposes, i.e., they were sufficient for proving a linear problem kernel on planar networks [2]. The following rules were basically introduced in [1] as a tool in the theoretical analysis of a search tree algorithm for DOMINATING SET on planar graphs. Notably, they lead to significant further improvements and speedups in our experimental analysis to follow.

1. *Delete edges between white vertices.*
2. *Let u be a white vertex of degree at most 1. Then, delete u.*
3. *Let u be a white vertex of degree 2, with two black neighbors u_1 and u_2.*
   - *(a) If u_1 and u_2 are connected by an edge, then delete u.*
   - *(b) If u_1 and u_2 are connected via a third (black or white) vertex u_3, then delete u.*
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.*

[2] 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.

	AS model: Oregon				refined AS model: Oregon+			
date	vertices	edges	% reduced	DS	vertices	edges	% reduced	DS
03/31/01	10670	22002	100%	957	10900	31180	100.00%	936
04/07/01	10729	21999	100%	969	10981	30855	99.97%	935
04/14/01	10790	22469	100%	978	11019	31761	99.92%	949
04/21/01	10895	22747	100%	982	11080	31538	99.95%	956
04/28/01	10886	22493	100%	991	11113	31434	100.00%	965
05/05/01	10943	22607	100%	988	11157	30943	99.89%	960
05/12/01	11011	22677	100%	988	11260	31303	99.89%	961
05/19/01	11051	22724	100%	979	11375	32287	99.90%	968
05/26/01	11174	23409	100%	993	11461	32730	99.92%	966

Table 1: **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 refined model ("Oregon+") by Chen *et al.* [3]. The columns show the size of the different networks, 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.

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 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 $G' = (B \uplus W, E)$ as follows: $B := \{u' \mid u \in V\}$, $W := \{u'' \mid u \in V\}$, and $E := \big\{\{u', u''\} \mid u \in V\big\} \cup \big\{\{u'', v'\}, \{u'', v''\} \mid (u, v) \in A\big\}$. 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 edges connecting u'' with u' and u''' 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.

3 Experimental Results

We tested our algorithmic framework on various network data provided in the literature and publically available on the web. Our focus was on networks obtained from (Internet) topology generators (Inet, BRITE) [8, 9], networks of autonomous systems [3]. Besides many others, one possible interest in computing small dominating sets in Internet networks might be time servers (NTP protocol). Here, the time servers (i.e., the vertices of a dominating set) might quickly provide the time signal to all other vertices in the network since following only one single link is enough. Finally, we will have a brief look at three examples of directed networks (an HTML network and two food web networks from biology). All our experiments ran on a 2.26 GHz Linux Pentium 4 PC with 1 GB main memory. The code was implemented in C++ using the algorithm library LEDA.

Autonomous systems networks. Chen *et al.* [3] 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, argue why these may provide an incomplete picture of the physical connectivity that exists in the actual Internet, and present a network model and refined connectivity maps that are supposed to provide a more complete picture of the Internet connectivity (see [3] 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 by Chen *et al.* We took both these 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, we were left with a drastically reduced network where one could easily compute the remaining optimal domination vertices by brute-force methods. Table 1 lists the results for the old model (here the computation per network took several minutes) and the

	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	21	100%	1085	13811	47	100%	1650	18532	93	100%	2129
d: 0.3	10434	148	100%	1062	15765	237	100%	1584	21145	386	100%	2102
d: 0.2	11084	250	100%	993	16758	467	100%	1483	22451	1338	100%	1955
d: 0.1	11470	888	100%	900	17733	1581	100%	1356	23764	4505	100%	1802
d: 0.05	12066	1758	100%	847	18225	4840	100%	1265	24416	10427	100%	1699
d: 0.001	12383	7373	99.9%	814	18712	14688	100%	1198	25045	27920	100%	1595

Table 2: **Inet 2.0 Topology Generator:** The table summarizes the performance of the data reduction on various networks generated with the generator in [8]. 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 [8] 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 (in seconds), the amount by which the networks were reduced, and the size of an optimal dominating set (DS) as computed by our method.

	BRITE: 1000 vertices, 1997 edges					BRITE: 5000 vertices, 9997 edges				
	Type 1	Type 2	Type 3	Type 4	Type 5	Type 1	Type 2	Type 3	Type 4	Type 5
# vertices removed (percentage)	1000 100%	668 64.8%	906 90.6%	915 91.5%	751 75.1%	4993 99.9%	4907 98.1%	4156 83.1%	4321 86.4%	4917 98.3%
# edges removed (percentage)	1997 100%	1450 72.6%	1873 93.8%	1892 94.7%	1558 78.0%	9990 99.9%	9893 98.9%	8569 85.7%	8982 89.9%	9907 99.1%
#vertices for DS found	195	120	173	176	146	941	934	771	792	913
time (sec)	77	127	67	71	87	5516	5472	7423	8496	8153

Table 3: **BRITE Topology Generator:** The table summarizes the performance of the data reduction on various networks generated with the generator in [9]. We constructed networks of 1000 and 5000 vertices using various parameter settings in the generator (Type 1–5). The parameters provided by BRITE are Node Placement Strategy (NPS), Growth Type (GT), Preference Connectivity (PF) (see [9] for details). We used the following settings: Type 1 (NPS:random, GT:all, PC:on), Type 2 (NPS:heavy, GT:incremental, PC:none), Type 3 (NPS:heavy, GT:incremental, PC:on), Type 4 (NPS:random, GT:all, PC:on), Type 5 (NPS:random, GT:incremental, PC:on). Each class consists of a sample of five networks and, for each network class, we averaged the number of vertices and edges removed by the reduction rules and the number of vertices that were determined to belong to an optimal dominating set (which also gives a lower bound on the size of a minimum dominating set of the overall network).

new model (here the computation per network took few hours).

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.

Networks from topology generators. Here we report on results using network data produced by the Internet topology generators Inet [8] and BRITE [9]. We refer to the given papers for any details. Tables 2 and 3 give our results and the parameter settings we used for generating the corresponding networks from Inet 2.0 and BRITE (Barabási-Albert model).

We only particularly emphasize few of our experimental findings. Concerning Inet networks, 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. The spectrum of running times is fairly big, ranging from about 20 seconds up to seven hours. The dominating set sizes, however, did not vary that much and were between more than 800 and less than 2130. Concerning BRITE networks, our rules were not quite as successful as in the case of Inet. For networks with 1000 and 5000 vertices, we had running times in the same dimension as for Inet, but more often an optimal dominating set could not completely be found by sole use of our rules. In particular, there were two network instances of Type 5 for 1000 vertices and Type 3 for 5000 vertices where our rules only achieved a small reduction of the network. By way of contrast, most networks of Type 3 and Type 5 (both for 1000 and 5000 vertices), which are considered to be particularly realistic models of the Internet [9], were almost always completely resolved.

Some directed networks. Finally, to gain first insights for directed networks, we also tested our rules on the proposed translation of directed networks into undirected ones. Because of the lack of space, however, we only mention results for three particular networks. Firstly, we considered an HTML network with 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`. Links in this document translated into arcs and particular pages translated into vertices. Within less than 10 seconds our reduction rules computed an optimal dominating set of size 141. 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 slightly more than one second an optimal dominating set of only 24 preys was determined. The second food web we tested is Ythan Estuary consisting of 270 vertices and 1286 arcs. In this case, after about 7 seconds we obtained 14 preys that are part of an optimal dominating set. We were left with a reduced network of 21 vertices and 43 arcs where no more reduction rule applied. Within few more seconds, using a tree decomposition based algorithm 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. 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 (also on social networks as discussed in [12]) remain to be conducted in future work.

Final Conclusion and Outlook. We demonstrated the potential of comparatively simple and easy to implement efficient 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 must for everyone dealing with domination in networks. On the "negative" side, our data reduction rules seem to behave poor when applied to dense graphs with many edges. Sanchis [11] generates these sorts of data and proposes heuristic algorithms to compute not necessarily optimal dominating sets in these settings. However, for many naturally occurring networks our data reduction rules performed extremely good.

References

[1] J. Alber, H. Fan, M. R. Fellows, H. Fernau, R. Niedermeier, F. Rosamond, and U. Stege. Refined search tree technique for Dominating Set on planar graphs. In *Proceedings 26th MFCS 2001*, Springer-Verlag LNCS 2136, pp. 111-122, 2001.

[2] J. Alber, M. R. Fellows, and R. Niedermeier. Efficient data reduction for Dominating Set: a linear problem kernel for the planar case. In *Proceedings 8th SWAT 2002*, Springer-Verlag LNCS 2368, pp. 150-159, 2002.

[3] Q. Chen, H. Chang, R. Govindam, S. Jamin, S. J. Shenker, and W. Willinger. The origin of power laws in internet topology revisited. In *Proceedings IEEE INFOCOM 2002.*

[4] U. Feige. A threshold of $\ln n$ for approximating Set Cover. *Journal of the ACM*, **45**:634–652, 1998.

[5] T. W. Haynes, S. M. Hedetniemi, S. T. Hedetniemi, and M. A. Henning. Domination in graphs applied to electric power networks. *SIAM J. Discrete Math.*, **15(4)**:519–529, 2002.

[6] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater. *Fundamentals of Domination in Graphs*. Monographs and Textbooks in Pure and Applied Mathematics Vol. 208, Marcel Dekker, 1998.

[7] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater. *Domination in Graphs*. Monographs and Textbooks in Pure and Applied Mathematics Vol. 209, Marcel Dekker, 1998.

[8] C. Jin and Q. Chen and S. Jamin. Inet: internet topology generator. Technical Report CSE-TR443-00, Department of EECS, University of Michigan, 2000.

[9] A. Medina, A. Lakhina, I. Matta, and J. Byers. BRITE: An approach to universal topology generation. In *Proceedings of MASCOTS 2001*, IEEE Computer Society, August 2001.

[10] F. S. Roberts. *Graph Theory and Its Applications to Problems of Society.* SIAM Press 1978. Third printing 1993 by Odyssey Press.

[11] L. A. Sanchis. Experimental analysis of heuristic algorithms for the Dominating Set problem. *Algorithmica*, **33**:3–18, 2002.

[12] T. W. Valente, B. Hoffman, A. Ritt-Olson, K. Lichtman, and C. A. Johnson. Use of network analysis to structure health promotion programs. Manuscript, School of Medicine, University of Southern California, August 2002.

[13] P.-J. Wan, K. M. Alzoubi, and O. Frieder. A simple heuristic for Minimum Connected Dominating Set in graphs. *International Journal of Foundations of Computer Science*, **14(2)**:323–333, 2003.