The Buss Reduction for the k-Weighted Vertex Cover Problem

Hong Xu Xin-Zeng Wu Cheng Cheng Sven Koenig T. K. Satish Kumar
University of Southern California, Los Angeles, California 90089, the United States of America
{hongx,xinzengw,chen260,skoenig}@usc.edu tkskwork@gmail.com

Abstract

The k-vertex cover (k-VC) problem is to find a VC of cardinality no more than k on a given undirected graph, and the k-weighted VC (k-WVC) problem is to find a VC of a weight no more than k on a given vertex-weighted undirected graph. In this paper, we generalize the Buss reduction, an important kernelization technique for the k-VC problem, to the k-WVC problem. We study its properties for the k-VC problem and the k-WVC problem on surrogates of large real-world graphs that are generated using the Erdős-Rényi model and the Barabási-Albert model. We also argue that our study of the Buss reduction bears important implications on the kernelization of combinatorial problems that have been shown to be reducible to WVC problems.

Introduction

Many interesting combinatorial problems in constraint reasoning, probabilistic reasoning, machine learning, operations research, and other research areas, are NP-hard. Despite many sophisticated search algorithms dedicated to solving them, the search spaces still remain intractable for large instances. Therefore, a polynomial-time procedure that reduces the sizes of problem instances and identifies a combinatorial core can be beneficial as a preprocessing step. Such a procedure is called a kernelization procedure, and the combinatorial core is called a kernel.

Despite its importance, kernelization is understudied for general combinatorial problems. Recent work presents a kernelization method for combinatorial problems posed as weighted constraint satisfaction problems (WCSPs) (Xu, Kumar, and Koenig 2017). This method relies on first reducing WCSPs to weighted vertex cover (weighted VC, aka WVC) problems and then using kernelization methods for the latter. A study of kernelization methods for (weighted) VC problems is therefore important towards understanding kernelization methods for general combinatorial problems.

(Weighted) VC problems have also been widely used to study various other real-world and theoretical problems. For example, they have been used in computer network security (Filiol et al. 2007), in crew scheduling (Sherali and Rios 1984), and in the construction of phylogenetic trees (Abu-Khzam et al. 2004). They have been used to prove the NP-completeness of various problems, such as the set cover problem and the dominating set problem (Korte and Vygen 2012). WVC problems have also been used to identify tractable subclasses of WCSPs using constraint composite graphs (Kumar 2008a; 2008b; 2016).

Formally, the following problems are defined on an undirected graph $G = \langle V, E\rangle$. A VC of G is defined as a set of vertices $S \subseteq V$ such that every edge in E has at least one of its endpoint vertices in S. The k-VC problem is to compute a VC of G with cardinality no more than k. If we assign a non-negative weight $w(v_i)$ to each vertex $v_i \in V$ of G, then G is called vertex-weighted and is denoted by $G = \langle V, E, w\rangle$. The k-WVC problem on G is to find a VC of total weight no more than k. The k-VC and k-WVC problems are NP-complete (Karp 1972).

A general direction for tackling (weighted) VC problems is to use kernelization. Kernelization reduces the number of variables before exhaustive search is initiated. Kernelization methods, such as the Nemhauser-Trotter reduction (Nemhauser and Trotter 1975) and the crown reduction (Abu-Khzam et al. 2007), are known for VC problems as well as WVC problems. However, these kernelization methods have been experimentally studied only for VC problems so far (Abu-Khzam et al. 2004; Díaz, Petit, and Thilikos 2006). The Buss reduction (due to S. R. Buss cited in (Buss and Goldsmith 1993)) is a well-known kernelization method for VC problems. It has been experimentally studied for the k-VC problem on random graphs generated with a preferential attachment model (Díaz, Petit, and Thilikos 2006).

Random graph models have been used to model real-world scenarios. For example, the Erdős-Rényi (ER) model (Erdős and Rényi 1959) has been used to model social and technological networks (Lerman, Yan, and Wu 2016; Wu, Percus, and Lerman 2017), and the Barabási-Albert (BA) model (Barabási and Albert 1999) has been used to model software package dependency networks (Horváth 2012). A study of kernelization methods on ER and BA graphs therefore has implications for their effectiveness and properties on real-world graphs.

In this paper, we first generalize the Buss reduction to the k-WVC problem. We then empirically study its properties for the k-VC problem and the k-WVC problem on random graphs generated using the ER and BA models. We make several interesting observations and substantiate them using theoretical arguments.

Algorithm 1: Buss reduction for the k-WVC problem

```
Function Buss ($G = \langle V, E, w \rangle$, $k$, $VC$)
    Input: $G$: A vertex-weighted undirected graph to
           find a $k$-WVC for.
    Input: $k$: The maximum total weight of the VC.
    Input and Output: $VC$: Vertices in the vertex
                      cover.
    Output: $K$: A Buss reduction kernel.
    Output: $S$: Status, i.e., what the Buss reduction
            concludes about the existence of a
            $k$-WVC.
    if $k < 0$ then
        return $K = \emptyset, S = NO$;
    else if $E = \emptyset$ then
        return $K = \emptyset, S = YES$;
    else if $\exists v \in V : \sum_{u \in \partial v} w(u) > k$ then
        $VC := VC \cup \{v\}$;
        $G' :=$ subgraph of $G$ induced by $V \setminus \{v\}$;
        return Buss ($G'$, $k - w(v)$, $VC$) ;
    else
        return $K = G \setminus I(G), S = UNKNOWN$;
```

The Buss Reduction for the k-WVC Problem

For the k-VC problem on an undirected graph $G = \langle V, E \rangle$, the Buss reduction works as follows (Díaz, Petit, and Thilikos 2006). It finds a vertex $v \in V$ with more than k adjacent vertices, adds v into the VC, and removes v from the graph. This results in a $(k-1)$-VC problem. It applies this procedure until no vertex can be removed or k vertices have been removed. The remaining graph is the Buss reduction kernel of the problem instance. The Buss reduction is correct, since, if v is not in the VC, then all of its adjacent vertices have to be in the VC to cover all the edges incident on v. This would make the size of the VC greater than k. The time complexity of the Buss reduction is $\mathcal{O}(k \cdot d \cdot |V|)$.

Here, we extend the Buss reduction to the k-WVC problem. On a vertex-weighted undirected graph $G = \langle V, E, w \rangle$, if there exists a vertex $v \in V$ such that the total weight of all its adjacent vertices is greater than k, then v must be selected in the VC. If it is not in the VC, then all adjacent vertices must be in the VC instead. This would result in a VC with a weight greater than k. The details of the Buss reduction are shown in Algorithm 1, where ∂v denotes the set of vertices adjacent to v and $I(G)$ denotes the set of isolated vertices in G, i.e., vertices of degree zero. If the graph is represented by an adjacency list, its time complexity is $\mathcal{O}(\frac{k}{w_m} \cdot d \cdot |V|)$, where $w_m = \min_{v \in V} w(v)$ and d is the maximum degree of all vertices. If the weights of all vertices in G are 1, Algorithm 1 is exactly the Buss reduction for the k-VC problem.

Random Graph Models

We study the Buss reduction on two random graph models: the ER model (Erdős and Rényi 1959) and the BA model (Barabási and Albert 1999). Both models are used to generate surrogates for large real-world graphs.

The ER model (Erdős and Rényi 1959) generates graphs in which each pair of vertices is connected by an edge with the same given probability. The ER model has been used to study real-world graphs, such as social and technological networks (Lerman, Yan, and Wu 2016; Wu, Percus, and Lerman 2017). Formally, the ER model has two parameters n and p. It generates a graph using these two parameters as follows: It first generates n vertices; then, for each each pair of vertices, it connects them with probability p. We define the connectivity parameter $c \equiv np$. Given this definition, we can equivalently describe an ER model using n and c.

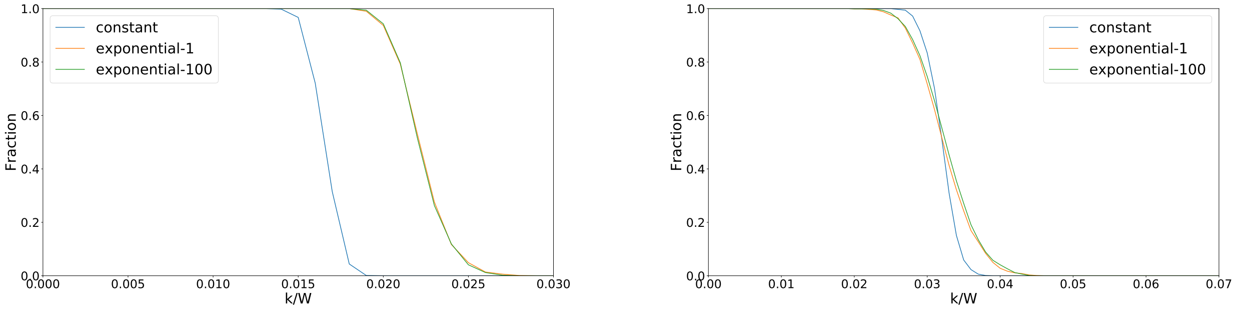
The BA model (Barabási and Albert 1999) uses a preferential attachment mechanism to generate random graphs whose vertices have degrees that follow the power law $P(k) \propto k^{-3}$. It has been widely used to study real-world graphs (Barabási 2016), such as software package dependency networks (Horváth 2012). Formally, it has three parameters n, m_0, and m. It generates a graph using these parameters as follows: It starts with a complete graph with $m_0 \geq 2$ vertices. It then adds new vertices, one at a time, until the graph has n vertices. When it adds a new vertex v, it connects v to m existing vertices. The probability with which v to an existing vertex u is proportional to the degree of u. In other words, v is attached to u with a preference that is linear in the degree of u.

Experimental Evaluation

In our experiments, we implemented the Buss reduction in C++. We generated two suites of instances. For each of these instance suites, we generated $1,000$ graphs with 100, 500 and $1,000$ vertices each. For the first instance suite, we generated the graphs using the ER model with $c = 8$. For the second instance suite, we generated the graphs using the BA model with $m = m_0 = 2$. For each graph in each of these instance suites, we generated 3 instances by setting all weights to the same constant 1, as well as according to exponential distributions $f_e(x; \lambda) = \lambda \exp(-\lambda x)$ with rate parameters $\lambda = 1$ and $\lambda = 100$. These distributions are referred to as "constant", "exponential-1" and "exponential-100", respectively. Setting all weights to be the constant 1 can be viewed as following an exponential distribution with rate parameter $\lambda = +\infty$[1]: They both have zero variance.

We first study the status output of the Buss reduction. No instance had an output of "YES". This means that the Buss reduction did not reduce any instance to an empty kernel if a k-WVC exists. Figure 1 shows the fraction of instances η that resulted in output "NO" for different vertex weight distributions. The remaining fraction resulted in output "UNKNOWN". As k increases, there is a sudden drop in η within a critical range of k. For the ER instances, as the rate parameter increases to $+\infty$, the critical range of k shifts to the left. For the BA instances, as the rate parameter increases to $+\infty$, the critical range of k becomes narrower, i.e., the drop becomes steeper. In both cases, the observed behavior of exponential-100 weights is very close to that of exponential-

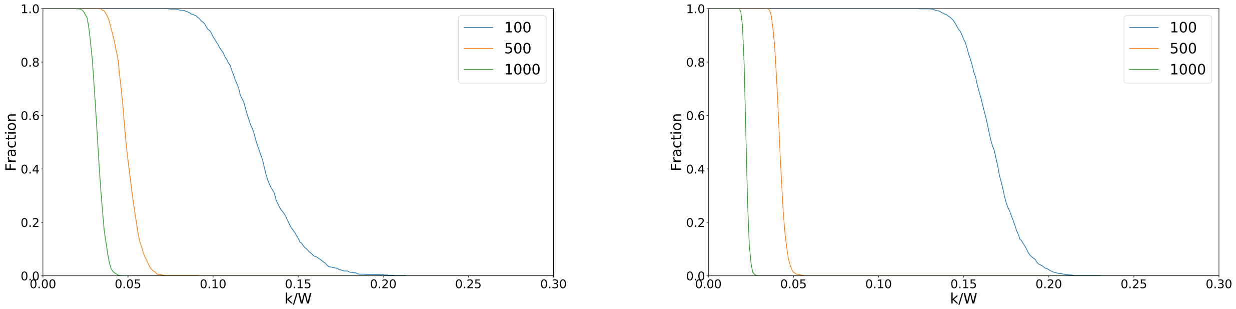
[1] Mathematically an exponential distribution with rate parameter $\lambda = +\infty$ is not defined. Here, we use $+\infty$ to represent extremely large real numbers.



(a) The ER Instances

(b) The BA Instances

Figure 1: Shows the fraction of instances η for which the Buss reduction outputs “NO” versus k/W for different weight distributions, where W is the total weight of the vertices in the graph. Only instances with 1,000 vertices are used here. The blue, orange, and green curves represent graphs that have constant, exponential-1, and exponential-100 weights, respectively.



(a) The ER Instances

(b) The BA Instances

Figure 2: Shows the fraction of instances η for which the Buss reduction outputs “NO” versus k/W for graphs of different sizes. Only instances with exponential-1 weights are used here. The blue, orange, and green curves represent graphs that have 100, 500, and 1,000 vertices, respectively.

1 weights. This is surprising since exponential-100 weights have a low variance of 10^{-4}. Therefore, its curve was expected to be closer to that of the constant weight.

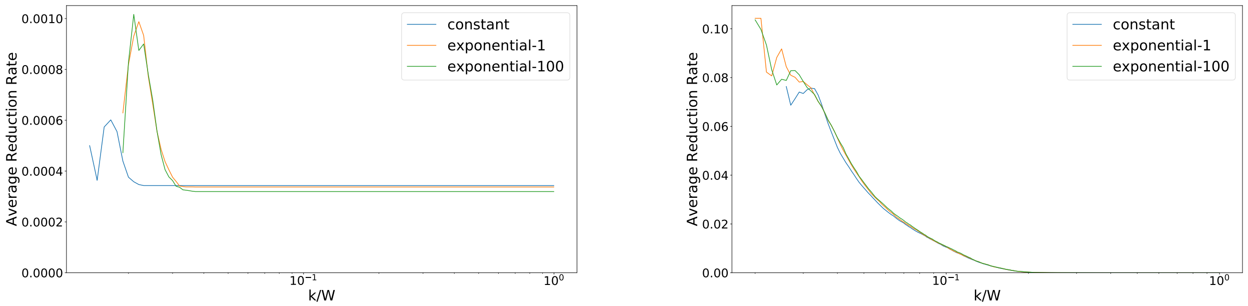
Figure 2 shows the fraction of instances η that output “NO” for different graph sizes. As the number of vertices increases, the critical range of k narrows and moves to the left. The narrowing may be explained using the argument that random graphs of infinite size commonly exhibit asymptotic properties more clearly than their finite counterparts: The output of the Buss reduction becomes more certain for every k as the graph size increases. The moving to the left may be explained by the fact that the number of vertices that can be reduced by the Buss reduction increases only sublinearly in the graph size for both the ER and BA models.

In further experiments, we use the reduction rate and the component reduction rate to measure the effectiveness of the Buss reduction. The reduction rate is defined as the ratio of the total weight of the vertices removed by the Buss reduction and the total weight of all vertices of the input graph. The component reduction rate is defined as one minus the ratio of the number of vertices in the largest connected component of the kernel and the number of vertices in the input graph. The larger the rates are, the more effective the Buss reduction is. Each measurement has its own advantages: The reduction rate measures how much the problem size has been reduced, and the component reduction rate measures how the hardness of the problem has been reduced—the time complexity of solving the k-WVC problem is exponential in the size of the maximum connected component.

Figures 3 and 4 show the average reduction rates and the average component reduction rate over instances that output “UNKNOWN”. For the ER instances, both rates are almost zero when k/W is greater than a threshold. For the BA instances, both rates decrease when k/W increases.

Analysis

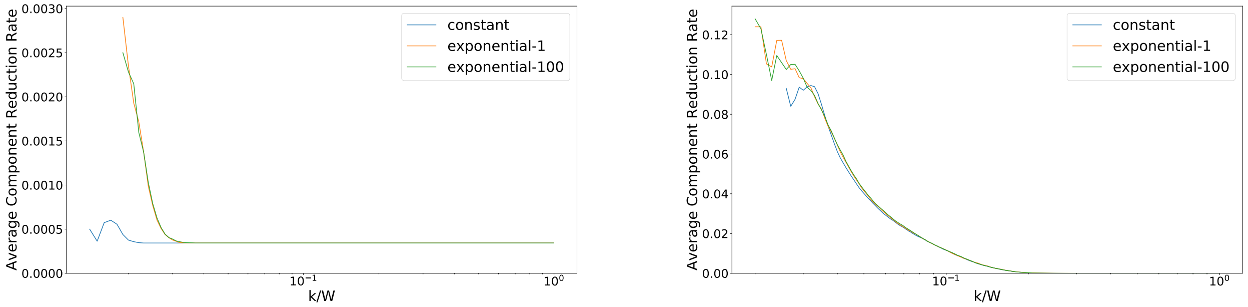
The shape of the curve of the reduction rates can be explained as follows. Let $F_d(\cdot)$ be the cumulative distribution function (CDF) of the degrees of vertices. For the k_0-VC problem, let the Buss reduction stop after removing $k_1 \leq k_0$ vertices. For any k, if the Buss reduction finds a vertex to remove for the k-VC problem, it invokes the $(k-1)$-VC prob-



(a) The ER Instances

(b) The BA Instances

Figure 3: Shows the average reduction rate as a function of k/W for different weight distributions. Only instances with 1,000 vertices are used here. The blue, orange, and green curves represent graphs that have constant, exponential-1, and exponential-100 weights, respectively.



(a) The ER Instances

(b) The BA Instances

Figure 4: Shows the average component reduction rate as a function of k/W for different weight distributions. Only instances with 1,000 vertices are used here. The blue, orange, and green curves represent graphs that have constant, exponential-1, and exponential-100 weights, respectively.

lem. Therefore, the reduction rate is equal to $\frac{k_0 - k_1}{N}$, where N is the number of vertices. On the other hand, if the number of neighbors of vertices did not change from iteration to iteration, the reduction rate can also be approximated as $1 - F_d(k_1)$, since $F_d(k_1)$ is the fraction of vertices that have degrees less or equal to k_1. Therefore, we can estimate k_1 by solving

$$1 - F_d(k_1) = \frac{k_0 - k_1}{N} \tag{1}$$

for k_1. Figure 5 visualizes the solution of Equation (1). The intersection points of the solid and dashed curves represent the solutions of Equation (1) for the ER and BA graphs. If the value of k_0 increases, the dashed curve moves to the right and the reduction rate decreases, as shown in Figure 6. Now, we let N approach ∞ to study the asymptotic behavior. Infinite ER graphs have Poisson vertex degree distributions, and infinite BA graphs have power-law vertex degree distributions. Since power-law distributions have fatter tails than Poisson distributions, the reduction rate drops faster for ER graphs than for BA graphs. For large k_0's, the reduction rates become 0 for both kinds of graphs.

Our analysis can be generalized to the k-WVC problem. Let the total weight of the vertices adjacent to a vertex v be $\Omega(v) = \sum_{u \in \partial v} w(u)$. The CDF of $\Omega(v)$ is

$$F_\Omega(\omega) = \sum_{j=1}^{\infty} \left[f_d(j) \int_{w_1 + \cdots + w_j \leq \omega} \prod_{i=1}^{j} f_w(w_i) \, \mathrm{d}w_i \right], \tag{2}$$

where $f_d(\cdot)$ is the probability density function (PDF) of the degrees of vertices, and $f_w(\cdot)$ is the PDF of the weights. To be consistent with our instance suites, we set $f_w(\cdot)$ to be the PDF of an exponential distribution $f_e(\cdot; \lambda)$. The sum of exponential independent and identically distributed random variables has a Gamma distribution. Therefore, Equation (2) can be written as

$$F_\Omega(\omega) = \sum_{j=1}^{\infty} \left[f_d(j) \int_0^{\omega} \frac{\gamma(j, \lambda t)}{(j-1)!} \, \mathrm{d}t \right]. \tag{3}$$

Here, $\gamma(\cdot, \cdot)$ is the incomplete Gamma function. For the k_0-WVC problem, let the Buss reduction stop after removing vertices of total weight $k_1 \leq k_0$. An argument similar to the

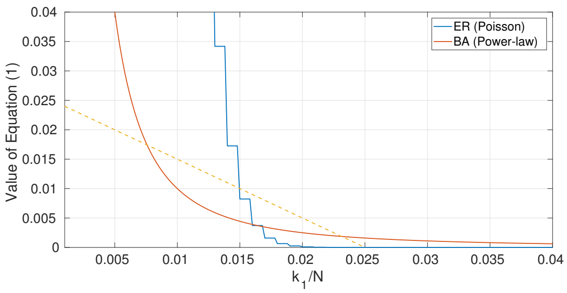


Figure 5: Shows the schematic plot for solving Equation (1) for some k_0. Here $N = 1,000$. The solid curves represent the left-hand side of Equation (1) of the ER and BA graphs, respectively. The dashed curve represent one possibility of the right-hand side of Equation (1).

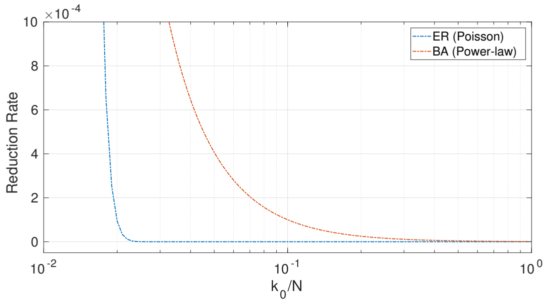


Figure 6: Shows the plots of the reduction rates as a function of k_0/N obtained by solving Equation (1) for $N = 1,000$ numerically for ER graphs with parameter $c = 8$ and BA graphs with parameters $m_0 = m = 2$.

k-VC problem shows that

$$1 - F_{\Omega}(k_1) = \frac{k_0 - k_1}{N\langle w \rangle}, \tag{4}$$

where $\langle w \rangle$ is the average vertex weight. Equation (4) is similar to Equation (1), admitting similar solutions to those shown in Figure 6. This is consistent with our experimental results.

Conclusions

We generalized the Buss reduction to the k-WVC problem. We then empirically studied its properties on two important random graph models: the ER model and the BA model. Experimentally, we showed that the Buss reduction typically does not reduce ER or BA graphs to empty kernels if a k-WVC exists. We showed that the fraction of instances for which the Buss reduction concludes the non-existence of a k-WVC drops significantly when k is within the critical range. We also showed that, by changing constant weights to weights sampled by exponential distributions, this critical range shifts to larger k's for the ER model and broadens for the BA model. We showed that, as the graph size increases, the critical range narrows and shifts to smaller k's. Further experiments on the reduction rates were substantiated with theoretical arguments. We showed that the reduction rate and the component reduction rate drop to near zero quickly for ER instances and more gradually for BA instances. Because the ER and BA models are characteristic of many real-world graphs and because many other combinatorial problems are reducible to (weighted) VC problems, our study of the Buss reduction has broader implications on kernelization of combinatorial problems.

Acknowledgment

The research at the University of Southern California was supported by the National Science Foundation (NSF) under grant numbers 1724392, 1409987, and 1319966. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the sponsoring organizations, agencies or the U.S. government.

References

Abu-Khzam, F. N.; Collins, R. L.; Fellows, M. R.; Langston, M. A.; Suters, W. H.; and Symons, C. T. 2004. Kernelization algorithms for the vertex cover problem: Theory and experiments. In *the Workshop on Algorithm Engineering and Experiments*.

Abu-Khzam, F. N.; Fellows, M. R.; Langston, M. A.; and Suters, W. H. 2007. Crown structures for vertex cover kernelization. *Theory of Computing Systems* 41(3):411–430.

Barabási, A.-L., and Albert, R. 1999. Emergence of scaling in random networks. *Science* 286(5439):509–512.

Barabási, A.-L. 2016. *Network Science*. Cambridge University Press, 1st edition.

Buss, J. F., and Goldsmith, J. 1993. Nondeterminism within P^*. *SIAM Journal on Computing* 22(3):560–572.

Díaz, J.; Petit, J.; and Thilikos, D. M. 2006. Kernels for the vertex cover problem on the preferred attachment model. In *the International Workshop on Experimental Algorithms*, 231–240.

Erdős, P., and Rényi, A. 1959. On random graphs I. *Publicationes Mathematicae* 6:290–297.

Filiol, E.; Franc, E.; Gubbioli, A.; Moquet, B.; and Roblot, G. 2007. Combinatorial optimisation of worm propagation on an unknown network. *International Journal of Computer, Electrical, Automation, Control and Information Engineering* 1(10):2931–2937.

Horváth, Á. 2012. The software package dependency networks of some Linux distributions. In *IEEE International Conference on Nonlinear Science and Complexity*, 235–238.

Karp, R. M. 1972. Reducibility among combinatorial problems. In *Complexity of Computer Computations*. Plenum Press, New York. 85–103.

Korte, B., and Vygen, J. 2012. *Combinatorial Optimization: Theory and Algorithms*. Springer, 5th edition.

Kumar, T. K. S. 2008a. A framework for hybrid tractability results in Boolean weighted constraint satisfaction problems. In *the International Conference on Principles and Practice of Constraint Programming*, 282–297.

Kumar, T. K. S. 2008b. Lifting techniques for weighted constraint satisfaction problems. In *the International Symposium on Artificial Intelligence and Mathematics*.

Kumar, T. K. S. 2016. Kernelization, generation of bounds, and the scope of incremental computation for weighted constraint satisfaction problems. In *the International Symposium on Artificial Intelligence and Mathematics*.

Lerman, K.; Yan, X.; and Wu, X.-Z. 2016. The “majority illusion” in social networks. *PLoS ONE* 11(2).

Nemhauser, G. L., and Trotter, L. E. 1975. Vertex packings: Structural properties and algorithms. *Mathematical Programming* 8(1):232–248.

Sherali, H. D., and Rios, M. 1984. An air force crew allocation and scheduling problem. *The Journal of the Operational Research Society* 35(2):91–103.

Wu, X.-Z.; Percus, A. G.; and Lerman, K. 2017. Neighbor-neighbor correlations explain measurement bias in networks. *Scientific Reports* 7(1):5576.

Xu, H.; Kumar, T. K. S.; and Koenig, S. 2017. The Nemhauser-Trotter reduction and lifted message passing for the weighted CSP. In *the International Conference on Integration of Artificial Intelligence and Operations Research Techniques in Constraint Programming*, 387–402.