Towards Understanding the Min-Sum Message Passing Algorithm for the Minimum Weighted Vertex Cover Problem: An Analytical Approach

Masaru Nakajima* **Hong Xu*** **Sven Koenig** **T. K. Satish Kumar**
University of Southern California, Los Angeles, California 90089, the United States of America
{masarun,hongx,skoenig}@usc.edu tkskwork@gmail.com

Abstract

Given a vertex-weighted undirected graph $G = \langle V, E, w \rangle$, the minimum weighted vertex cover (MWVC) problem is to find a subset of vertices with minimum total weight such that every edge in the graph has at least one of its endpoints in it. The MWVC problem and its amenability to the min-sum message passing (MSMP) algorithm remain understudied despite the common occurrence of the MWVC problem and the common use of the MSMP algorithm in many areas of AI. In this paper, we first develop the MSMP algorithm for the MWVC problem that can be viewed as a generalization of the warning propagation algorithm. We then study properties of the MSMP algorithm for the MWVC problem on a special class of graphs, namely single loops. We compare our analytical results with experimental observations and argue that: (a) Our analytical framework is powerful in accurately predicting the behavior of the MSMP algorithm on the MWVC problem, and (b) for a given combinatorial optimization problem, it may be more effective to apply the MSMP algorithm on the MWVC problem that is equivalent to the given problem, instead of applying the MSMP algorithm on the given problem directly.

Introduction

Given an undirected graph $G = \langle V, E \rangle$, a *vertex cover* (VC) of G is defined as a subset of vertices $S \subseteq V$ such that every edge in E has at least one of its endpoint vertices in S. A *minimum vertex cover* (MVC) of G is a vertex cover of minimum cardinality. When G is vertex-weighted—i.e., each vertex $v_i \in V$ has a non-negative weight w_i associated with it—the *minimum weighted vertex cover* (MWVC) for it is defined as a vertex cover of minimum total weight. The MVC/MWVC problem is to find an MVC/MWVC.

Two important combinatorial optimization problems equivalent to the MVC problem are the *maximum independent set* (MIS) problem and the *maximum clique* (MC) problem (Cormen et al. 2009). The MVC problem and its equivalent MIS and MC problems have numerous real-world applications such as in AI scheduling, logistics and operations management, and VLSI design (Cai et al. 2013). More recent applications have also been discovered in information retrieval, signal processing, and sequence alignment in computational genomics (Johnson and Trick 1996).

*Masaru Nakajima and Hong Xu are both first authors.

Since the MVC problem is a special case of the MWVC problem, the latter not only captures all of the real-world combinatorial optimization problems that the MVC problem can model but also captures a wide range of additional combinatorial optimization problems central to AI. For example, consider a simple combinatorial auction problem (Sandholm 2002). We are given a set of items with bids placed on subsets of the items. Each bid has a valuation. The goal is to pick a set of winning bids that maximizes the total valuation—i.e., the revenue of the auctioneer—such that the set of items of the winning bids are pairwise disjoint. This can be modeled as the *maximum weighted independent set* (MWIS) problem—equivalent to the MWVC problem—on a graph constructed as follows: We create a vertex for each bid such that the weight of the vertex is equal to the valuation of that bid. Two vertices are connected by an edge if and only if their corresponding bids have a non-empty intersection. The winning bids correspond to the vertices in the MWIS of the graph.

While there are some reasonably good solvers for the MVC problem, the MWVC problem remains understudied. Clearly, the MWVC problem, as a generalization of the MVC problem, is harder to solve efficiently. Exact algorithms (Niskanen and Östergård 2003; Xu, Kumar, and Koenig 2016) are not expected to do well for large instances of the MWVC problem simply because they do not scale well even for large instances of the MVC problem. Moreover, the local search techniques used in the best solvers for the MVC problem are also not expected to generalize well to the MWVC problem because the MVC problem is fixed-parameter tractable while the MWVC problem is not (Chen, Kanj, and Xia 2006). The local search solvers for the MVC problem (Richter, Helmert, and Gretton 2007; Cai et al. 2013) heavily rely on this property as they solve the fixed-parameter vertex cover problem in their inner loops.

The MWVC problem is not only known to be hard, but is also understudied for its amenability to many popular algorithmic techniques. One such widely used technique is *message passing*. The *min-sum message passing* (MSMP) algorithm, a special type of the message passing algorithm, is a well known technique for solving many combinatorial optimization problems across a wide range of fields, such as probabilistic reasoning, artificial intelligence, statistical physics, and information theory (Mézard and Montanari

Citation: Masaru Nakajima, Hong Xu, Sven Koenig, and T. K. Satish Kumar. Towards understanding the min-sum message passing algorithm for the minimum weighted vertex cover problem: An analytical approach. In Proceedings of the 15th International Symposium on Artificial Intelligence and Mathematics (ISAIM). 2018.

2009; Yedidia, Freeman, and Weiss 2003). It is based on local information processing and communication, and avoids an exponential time complexity with respect to the size of the problem. It works well in practice on many combinatorial optimization problems such as those that arise in statistical physics, computer vision, error-correcting coding theory, or, more generally, on graphical models such as Bayesian networks and Markov random fields (Yedidia, Freeman, and Weiss 2003). It has also been used to study problems such as K-satisfiability (Mézard and Zecchina 2002) and the *weighted constraint satisfaction problem* (WCSP) (Xu, Kumar, and Koenig 2017). Although the MSMP algorithm convergences and guarantees correctness when the variable interactions form a tree, a complete theoretical analysis of its convergence and correctness for the general case is elusive.

Despite the individual importance of the MWVC problem and the MSMP algorithm, a detailed study of the effectiveness and properties of the MSMP algorithm applied to the MWVC problem has been missing. In this paper, we first develop the MSMP algorithm for the MWVC problem that can be viewed as a generalization of the *warning propagation* (WP) algorithm. We then study properties of the MSMP algorithm for the MWVC problem. We compare our analytical results with experimental observations, and argue that: (a) Our analytical framework is powerful in accurately predicting the behavior of the MSMP algorithm on the MWVC problem, and (b) for a given combinatorial optimization problem, it may be more effective to apply the MSMP algorithm on the MWVC problem that is equivalent to the given problem, instead of applying the MSMP algorithm on the given problem directly.

Related Work

(Weigt and Zhou 2006) derive the sizes of MVCs on infinite Erdös-Rényi (ER) random graphs (Erdős and Rényi 1959) by using the WP algorithm. They show that, on an infinite ER random graph with connectivity $c < e$, the size of an MVC is $1 - \frac{W(c)}{c} - \frac{W(c)^2}{2c}$, where $W(\cdot)$ is the Lambert-W function. Here, connectivity is the minimum number of elements (vertices or edges) that need to be removed to disconnect the remaining vertices from each other and e is the Euler's number. They derive this result using the convergence condition and equations for solution extraction of the WP algorithm. However, they do not mention any possible extension of their work to the MWVC problem.

Min-Sum Message Passing for the WCSP

We now review how the MSMP algorithm can be applied to solve the Boolean WCSP (Xu, Kumar, and Koenig 2017). This review is important for a proper explanation of the development of the MSMP algorithm on the MWVC problem. The WCSP is defined by a triplet $\langle \mathcal{X}, \mathcal{D}, \mathcal{C} \rangle$, where $\mathcal{X} = \{X_1, X_2, \ldots, X_N\}$ is a set of N variables, $\mathcal{D} = \{D(X_1), D(X_2), \ldots, D(X_N)\}$ is a set of N domains with discrete values, and $\mathcal{C} = \{C_1, C_2, \ldots, C_M\}$ is a set of M weighted constraints. Each variable $X_i \in \mathcal{X}$ can be assigned a value in its associated domain $D(X_i) \in \mathcal{D}$. Each constraint $C_i \in \mathcal{C}$ is defined over a subset of the variables $S(C_i) \subseteq \mathcal{X}$, called the scope of C_i. C_i associates a non-negative weight with each possible assignment of values to the variables in $S(C_i)$. The goal is to find a complete assignment of values to all variables in $\mathcal{X}$ from their respective domains that minimizes the sum of the weights specified by each constraint in $\mathcal{C}$ (Bistarelli et al. 1999). Such assignment is called an optimal solution. This combinatorial task can equivalently be characterized by having to compute $\arg\min_{a \in A(\mathcal{X})} \sum_{C_i \in \mathcal{C}} E_{C_i}(a|S(C_i))$, where $A(\mathcal{X})$ represents the set of all $|D(X_1)| \times |D(X_2)| \times \ldots \times |D(X_N)|$ complete assignments to all variables in $\mathcal{X}$. $a|S(C_i)$ represents the projection of a complete assignment a onto the subset of variables in $S(C_i)$. E_{C_i} is a constraint function that maps each $a|S(C_i)$ to its associated weight in C_i. The Boolean WCSP is the WCSP with only variables of domain size 2, i.e., $\forall X_i \in \mathcal{X} : |D(X_i)| = 2$. It is representationally as powerful as the WCSP.

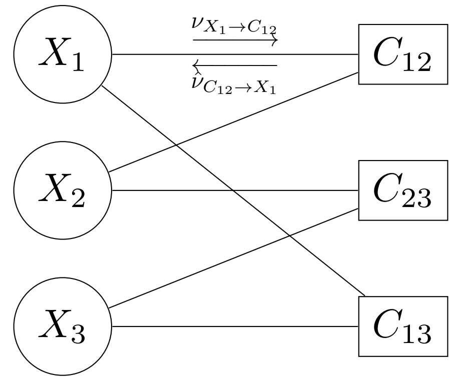

Figure 1: Illustrates the factor graph of a Boolean WCSP instance with 3 variables $\{X_1, X_2, X_3\}$ and 3 constraints $\{C_{12}, C_{13}, C_{23}\}$. Here, $X_1, X_2 \in S(C_{12})$, $X_1, X_3 \in S(C_{13})$ and $X_2, X_3 \in S(C_{23})$. The circles are variable vertices, and the squares are constraint vertices. $\nu_{X_1 \to C_{12}}$ and $\hat{\nu}_{C_{12} \to X_1}$ are the messages from X_1 to C_{12} and from C_{12} to X_1, respectively. Such a pair of messages annotates each edge (not all are explicitly shown).

To apply the MSMP algorithm to the Boolean WCSP, we first construct its factor graph. We create a vertex for each variable in $\mathcal{X}$ (variable vertex) and for each weighted constraint in $\mathcal{C}$ (constraint vertex). A variable vertex X_i and a constraint vertex C_j are connected by an edge if and only if $X_i \in S(C_j)$. Figure 1 shows an example.After the factor graph is constructed, a message (two real numbers) for each of the two directions along each edge is initialized, for instance, to zeros. A pair of messages $\nu_{X_1 \to C_{12}}$ and $\hat{\nu}_{C_{12} \to X_1}$ is illustrated in Figure 1. The messages are then updated iteratively by using the min-sum update rules given by

$$\nu_{X_i \to C_j}^{(t)}(x_i) = \sum_{C_k \in \partial X_i \setminus \{C_j\}} \left[\hat{\nu}_{C_k \to X_i}^{(t-1)}(x_i) \right] + c_{X_i \to C_j}^{(t)} \tag{1}$$

$$\hat{\nu}_{C_j \to X_i}^{(t)}(x_i) = \min_{a \in \mathcal{A}(\partial C_j \setminus \{X_i\})} \Bigg[E_{C_j}(a \cup \{X_i = x_i\}) + \sum_{X_k \in \partial C_j \setminus \{X_i\}} \nu_{X_k \to C_j}^{(t)}(a|\{X_k\}) \Bigg] + \hat{c}_{C_j \to X_i}^{(t)}, \tag{2}$$

for all $X_i \in \mathcal{X}, C_j \in \mathcal{C}$, $x_i \in \{0, 1\}$, and all $t > 0$ until convergence (Mézard and Montanari 2009), where

- $\hat{\nu}_{C_j \to X_i}^{(t)}(x_i)$ for both $x_i \in \{0, 1\}$ are the two real numbers of the message that is passed from constraint vertex C_j to variable vertex X_i in the t^{th} iteration,
- $\nu_{X_i \to C_j}^{(t)}(x_i)$ for both $x_i \in \{0, 1\}$ are the two real numbers of the message that is passed from variable vertex X_i to constraint vertex C_j in the t^{th} iteration,
- ∂X_i and ∂C_j are the sets of neighboring vertices of X_i and C_j, respectively, and
- $c_{X_i \to C_j}^{(t)}$ and $\hat{c}_{C_j \to X_i}^{(t)}$ are normalization constants such that

$$\min \left[\nu_{X_i \to C_j}^{(t)}(0), \nu_{X_i \to C_j}^{(t)}(1) \right] = 0 \tag{3}$$

$$\min \left[\hat{\nu}_{C_j \to X_i}^{(t)}(0), \hat{\nu}_{C_j \to X_i}^{(t)}(1) \right] = 0. \tag{4}$$

The message update rules can be understood as follows. Each message from a variable vertex X_i to a constraint vertex C_j is updated by summing up all of X_i's incoming messages from its other neighboring vertices. Each message from a constraint vertex C_j to a variable vertex X_i is updated by finding the minimum of the constraint function E_{C_j} plus the sum of all of C_j's incoming messages from its other neighboring vertices. The messages can be updated in various orders.

We remove the superscript (t) on messages to indicate the values of messages upon convergence. The final assignment of values to variables in $\mathcal{X} = \{X_1, X_2, \ldots, X_N\}$ is then found by computing

$$E_{X_i}(X_i = x_i) = \sum_{C_k \in \partial X_i} \hat{\nu}_{C_k \to X_i}(x_i) \tag{5}$$

for all $X_i \in \mathcal{X}$ and $x_i \in \{0, 1\}$. Here, $E_{X_i}(X_i = 0)$ and $E_{X_i}(X_i = 1)$ can be proven to be the minimum values of the total weights conditioned on $X_i = 0$ and $X_i = 1$, respectively. By selecting the value of x_i that leads to a smaller value of $E_{X_i}(X_i = x_i)$, we obtain the final assignment of values to all variables in $\mathcal{X}$.

The MSMP algorithm converges and produces an optimal solution if the factor graph is a tree. However, it is not necessarily the case if the factor graph is loopy (Mézard and Montanari 2009). Although the clique tree algorithm alleviates this problem to a certain extent by first converting loopy graphs to trees (Koller and Friedman 2009), the technique only scales to graphs with low treewidths. If the MSMP algorithm operates directly on loopy graphs, the theoretical underpinnings of its convergence and optimality properties still remain poorly understood.

In this context, our contribution is to provide the first analytical framework for a theoretical analysis of the MSMP algorithm for the MWVC problem with a loopy structure. Although our analysis is restricted to the MWVC problem, it provides a useful handle on the general case as well because the WCSP is reducible to the MWVC problem on its constraint composite graph (Kumar 2008a; 2008b; 2016).

Message Passing for the MWVC Problem

We first reformulate the MWVC problem as a subclass of the Boolean WCSP in order to make the MSMP algorithm applicable to it. Since this subclass of the Boolean WCSP contains only specific types of constraints, all equations used in the MSMP algorithm for the Boolean WCSP can be simplified for the MWVC problem. Here, we use an approach similar to (Xu, Kumar, and Koenig 2017) to derive these simplified message update equations. For notational convenience, we omit the normalization constants in the following derivation.

For an MWVC problem instance P on a graph $G = \langle V, E, w \rangle$, we associate a variable $X_i \in \{0, 1\}$ with each vertex $i \in V$. X_i represents the presence of i in the to-be-determined MWVC. P has two types of constraints:

- Unary weighted constraints: A unary weighted constraint corresponds to a vertex in G. We use C_i to denote the unary weighted constraint that corresponds to the vertex i. C_i therefore has only one variable X_i in its scope. In the weighted constraint C_i, the tuple in which $X_i = 1$ has weight $w_i \geq 0$ and the other tuple has weight zero. This type of weighted constraint represents the minimization objective of the MWVC problem. Formally, we have

$$E_{C_i}(X_i) = \begin{cases} w_i & \text{if} \quad X_i = 1 \\ 0 & \text{if} \quad X_i = 0 \end{cases}. \tag{6}$$

- Binary weighted constraints: A binary weighted constraint corresponds to an edge in G. We use C_{ij} to denote the binary weighted constraint that corresponds to the edge $\{i, j\}$. C_{ij} has two variable X_i and X_j in its scope. The tuple where $X_i = X_j = 0$ has weight infinity, and the other tuples have weight zero. This type of constraint represents the requirement that at least one endpoint vertex must be in the MWVC for each edge. Formally, we have

$$E_{C_{ij}}(X_i, X_j) = \begin{cases} +\infty & \text{if} \quad X_i = X_j = 0 \\ 0 & \text{if} \quad \text{otherwise} \end{cases}. \tag{7}$$

We build the factor graph G_P for P. Then, we have

$$\partial C_i \setminus \{X_i\} = \emptyset \tag{8}$$

$$\partial C_{ij} \setminus \{X_i\} = \{X_j\}. \tag{9}$$

By plugging Equations (6) and (8) into Equation (2), we have

$$\hat{\nu}_{C_i \to X_i}^{(t)}(x_i) = \begin{cases} w_i & \text{if} \quad x_i = 1 \\ 0 & \text{if} \quad x_i = 0 \end{cases} \tag{10}$$

for all C_i. Note that we do not need Equation (1) for C_i here since it has only one variable and thus the message passed to it does not affect the final solution.

By plugging Equations (7), (9) and (10) into Equations (1) and (2) along with the fact that there exist only unary and binary constraints, we have

$$\nu_{X_i \to C_{ij}}^{(t)}(1) = \sum_{C \in \partial X_i \setminus \{C_i, C_{ij}\}} \left[\hat{\nu}_{C \to X_i}^{(t-1)}(1)\right] + w_i \tag{11}$$

$$\nu_{X_i \to C_{ij}}^{(t)}(0) = \sum_{C \in \partial X_i \setminus \{C_i, C_{ij}\}} \hat{\nu}_{C \to X_i}^{(t-1)}(0) \tag{12}$$

$$\hat{\nu}_{C_{ij} \to X_i}^{(t)}(1) = \min_{b \in \{0,1\}} \nu_{X_j \to C_{ij}}^{(t)}(b) \tag{13}$$

$$\hat{\nu}_{C_{ij} \to X_i}^{(t)}(0) = \nu_{X_j \to C_{ij}}^{(t)}(1) \tag{14}$$

for all edges $\{i, j\}$. By plugging Equations (11) and (12) into Equations (13) and (14), we have

$$\hat{\nu}_{C_{ij} \to X_i}^{(t)}(1) = \min_{b \in \{0,1\}} \Bigg[\sum_{C \in \partial X_j \setminus \{C_j, C_{ij}\}} \left[\hat{\nu}_{C \to X_j}^{(t-1)}(b)\right] + w_j \cdot b \Bigg] \tag{15}$$

$$\hat{\nu}_{C_{ij} \to X_i}^{(t)}(0) = \sum_{C \in \partial X_j \setminus \{C_j, C_{ij}\}} \left[\hat{\nu}_{C \to X_j}^{(t-1)}(1)\right] + w_j \tag{16}$$

for all edges $\{i, j\}$, where $\hat{\nu}_{C_{ij} \to X_i}^{(t)}(b)$ for both $b \in \{0, 1\}$ are the two real numbers of the message that is passed from the edge $\{i, j\}$ to the vertex i. Since each edge has exactly two endpoint vertices, the message from an edge to one of its endpoint vertices can be viewed as a message from the other endpoint vertex to it. In addition, since we always normalize the messages during message passing (we omit the normalization constants in the equations above) by subtracting a number such that $\min_{b \in \{0,1\}} \left[\hat{\nu}_{C_{ij} \to X_i}^{(t)}(b)\right] = 0$, and $\hat{\nu}_{C_{ij} \to X_i}^{(t)}(1) \le \hat{\nu}_{C_{ij} \to X_i}^{(t)}(0)$ always holds, we only need to pass one number between adjacent vertices instead of two. Formally, we define the message from vertex i to vertex j in the t^{th} iteration as

$$\nu_{j \to i}^{(t)} \equiv \hat{\nu}_{C_{ij} \to X_i}^{(t)}(0) - \hat{\nu}_{C_{ij} \to X_i}^{(t)}(1), \tag{17}$$

for edge $\{i, j\}$. By plugging Equations (15) and (16) into Equation (17), we have the message update rules rewritten in the form of messages between vertices in G as

$$\begin{aligned} \nu_{j \to i}^{(t)} = & \sum_{C \in \partial X_j \setminus \{C_j, C_{ij}\}} \left[\hat{\nu}_{C \to X_j}^{(t-1)}(1)\right] + w_j \\ & - \min \left\{ \sum_{C \in \partial X_j \setminus \{C_j, C_{ij}\}} \left[\hat{\nu}_{C \to X_j}^{(t-1)}(0)\right], \right. \\ & \left. \sum_{C \in \partial X_j \setminus \{C_j, C_{ij}\}} \left[\hat{\nu}_{C \to X_j}^{(t-1)}(1)\right] + w_j \right\} \\ = & \max \\ & \left\{ w_j - \sum_{C \in \partial X_j \setminus \{C_j, C_{ij}\}} \left[\hat{\nu}_{C \to X_j}^{(t-1)}(0) - \hat{\nu}_{C \to X_j}^{(t-1)}(1)\right], 0 \right\} \\ = & \max \left\{ w_j - \sum_{k \in N(j) \setminus i} \nu_{k \to j}^{(t-1)}, 0 \right\} \end{aligned} \tag{18}$$

for edges $\{i, j\}$, where $N(j)$ is the set of neighboring vertices of j in G. Equation (18) is the message update rule of the MSMP algorithm adapted to the MWVC problem.

Using Equations (5) and (18), the decision of whether or not to include vertex i in the MWVC is made by calculating

$$\begin{aligned} & E_{X_i}(X_i = 0) - E_{X_i}(X_i = 1) \\ & = \sum_{C \in \partial X_i} \hat{\nu}_{C \to X_i}(0) - \sum_{C \in \partial X_i} \hat{\nu}_{C \to X_i}(1) \\ & = \sum_{C_{ij} \in \partial X_i} \left[\hat{\nu}_{C_{ij} \to X_i}(0) - \hat{\nu}_{C_{ij} \to X_i}(1)\right] - w_i \\ & = \sum_{j \in N(i)} \nu_{j \to i} - w_i. \end{aligned} \tag{19}$$

Equation (19) suggests that vertex i is in the MWVC if $w_i \le \sum_{j \in N(i)} \nu_{j \to i}$; vertex i is not in the MWVC if $w_i > \sum_{j \in N(i)} \nu_{j \to i}$. The case of $w_i = \sum_{j \in N(i)} \nu_{j \to i}$, however, is often extremely rare for many vertex weight distributions and can always be avoided in practice by perturbing the weights. Here, for theoretical analysis, we will select such a vertex i into the MWVC with probability $\frac{1}{2}$. Equation (18) is reduced to WP for the MVC problem if $\forall i \in V : w_i = 1$ (Weigt and Zhou 2006).

Here we argue by contradiction that the MSMP algorithm for the MWVC problem always outputs a VC if it converges. We assume that neither of the two adjacent vertices i and j is selected in the MWVC. Then, we have

$$\nu_{j \to i} + \nu_{\to i} < w_i \tag{20}$$

$$\nu_{i \to j} + \nu_{\to j} < w_j, \tag{21}$$

where $\nu_{\to i} = \sum_{k \in N(i) \setminus j} \nu_{k \to i}$ and $\nu_{\to j} = \sum_{k \in N(j) \setminus i} \nu_{k \to j}$. These two equations also imply that, Equation (18) for $\nu_{j \to i}$ and $\nu_{i \to j}$ can have the $\max$ operator removed. Then, we have

$$\nu_{j \to i} = w_j - \nu_{\to j} \tag{22}$$

$$\nu_{i \to j} = w_i - \nu_{\to i}. \tag{23}$$

By plugging Equations (22) and (23) into Equations (20) and (21), we have

$$w_j - \nu_{\to j} + \nu_{\to i} < w_i \tag{24}$$
$$w_i - \nu_{\to i} + \nu_{\to j} < w_j. \tag{25}$$

Adding these two equations, we have $w_i + w_j < w_i + w_j$, which is a contradiction.

The Probability Distribution of The Messages

We assume that the MWVC problem is posed on an infinitely large random graph that is generated according to a given random graph model. We assume that, upon convergence of the MSMP algorithm, the probability distribution of a message depends only on the weight of its sender. We use $f(\nu_{i\to j}; w_i)$ to denote the probability density function by which a vertex i with weight w_i sends the message $\nu_{i\to j}$ to its adjacent vertex j. $f(\nu_{i\to j}; w_i)$ can be calculated according to the joint probability distribution of the messages from all the adjacent vertices of i excluding j. Here, we use our approach for analyzing the behavior of the MSMP algorithm on graphs with a single loop (each vertex has exactly two adjacent vertices) with vertex weight distribution $g(w)$. For all $w_i \geq 0$, the cumulative probability function of $\nu_{i\to j}$ has the form

$$\begin{aligned} F(\nu_{i\to j}; w_i) = \Theta(\nu_{i\to j})P(0; w_i) + F_m(\nu_{i\to j}; w_i) \\ + \Theta(\nu_{i\to j} - w_i)P(w_i; w_i), \end{aligned} \tag{26}$$

where $P(0; w_i)$ and $P(w_i; w_i)$ are the probabilities of $\nu_{i\to j} = 0$ and $\nu_{i\to j} = w_i$, respectively; $F_m(\nu_{i\to j}; w_i)$ is assumed to be, with respect to $\nu_{i\to j}$, smooth in the interval $(0, w_i)$ and constant in $(-\infty, 0]$ and $[w_i, +\infty)$; and $\Theta(\cdot)$ is a step function

$$\Theta(x) = \begin{cases} 1, & \text{if} \quad x \geq 0 \\ 0, & \text{if} \quad x < 0. \end{cases} \tag{27}$$

Here, the first and third terms of Equation (26) are used to capture the step function-like behavior of the messages as evident in Equation (18) due to the effect of the $\max$ operator. By taking the derivative of $F(\nu_{i\to j}; w_i)$ with respect to $\nu_{i\to j}$, we have the probability density function of $\nu_{i\to j}$ of the form

$$\begin{aligned} f(\nu_{i\to j}; w_i) = \delta(\nu_{i\to j})P(0; w_i) + f_m(\nu_{i\to j}; w_i) \\ + \delta(\nu_{i\to j} - w_i)P(w_i; w_i), \end{aligned} \tag{28}$$

where $f_m(\nu_{i\to j}; w_i) = \partial F_m(\nu_{i\to j}; w_i)/\partial \nu_{i\to j}$, and $\delta(\cdot)$ is a delta function defined to satisfy

$$\begin{aligned} &\forall \text{ function } q(\cdot), \forall a < b : \\ &\int_a^b \mathrm{d}x\, q(x)\delta(x) = \begin{cases} q(0) & \text{if } b \geq 0 \wedge a < 0 \\ 0 & \text{otherwise} \end{cases}. \end{aligned} \tag{29}$$

We note that, based on the assumption on $F_m(\nu_{i\to j}; w_i)$, $f_m(\nu_{i\to j}; w_i)$ is, with respect to $\nu_{i\to j}$, a smooth function in the interval $(0, w_i)$ and 0 elsewhere.

We now find the expressions for $P(0; w_i)$, $f_m(\nu_{i\to j}; w_i)$, and $P(w_i; w_i)$. Equation (18) implies

$$\forall \{i, j\} \in E : \nu_{i\to j} \leq w_i. \tag{30}$$

Also, upon convergence, Equation (18) becomes

$$\nu_{i\to j} = \begin{cases} w_i & \text{if } \nu_{k\to i} = 0 \\ w_i - \nu_{k\to i} & \text{if } 0 < \nu_{k\to i} < w_i \\ 0 & \text{if } \nu_{k\to i} \geq w_i, \end{cases} \tag{31}$$

where vertex k vertex is adjacent to vertex i.

We first consider $f_m(\nu_{i\to j}; w_i)$. Since $f_m(\nu_{i\to j}; w_i) = 0$ for $\nu_{i\to j} \leq 0$ and $\nu_{i\to j} \geq w_i$, we focus on the range $0 < \nu_{i\to j} < w_i$. This condition corresponds to the second case of Equation (31), i.e., $\nu_{k\to i} = w_i - \nu_{i\to j}$. The probability density with which vertex k with weight ranging from w_k to $w_k + \mathrm{d}w_k$ sends the message $\nu_{k\to i} = w_i - \nu_{i\to j}$ is $\mathrm{d}w_k\ g(w_k)f(w_i - \nu_{i\to j}; w_k)$. Taking the integral over the weight distribution, we obtain

$$\begin{aligned} &f_m(\nu_{i\to j}; w_i) = \int_{(w_i-\nu_{i\to j})^-}^{+\infty} \mathrm{d}w_k\, g(w_k)f(w_i - \nu_{i\to j}; w_k) \\ &= \int_{(w_i-\nu_{i\to j})^-}^{+\infty} \mathrm{d}w_k\, g(w_k)\delta(w_i - \nu_{i\to j})P(0; w_k) \\ &+ \int_{w_i-\nu_{i\to j}}^{+\infty} \mathrm{d}w_k\, g(w_k)f_m(w_i - \nu_{i\to j}; w_k) \\ &+ \int_{(w_i-\nu_{i\to j})^-}^{+\infty} \mathrm{d}w_k\, g(w_k)\delta(w_i - \nu_{i\to j} - w_k)P(w_k; w_k) \end{aligned} \tag{32}$$

for $0 < \nu_{i\to j} < w_i$, where the lower integration limit is imposed by Equation (30), i.e., $w_k \geq \nu_{k\to i} = w_i - \nu_{i\to j}$, and $\int_{(w_i-\nu_{i\to j})^-}$ is short for $\lim_{\epsilon\to 0^+} \int_{(w_i-\nu_{i\to j})-\epsilon}$. The first term vanishes, since in the Delta function, $w_i - \nu_{i\to j} > 0$.

In order to analyze $P(0; w_i)$, which corresponds to the third case of Equation (31), we assume that $\nu_{k\to i} \geq w_i$. This inequality imposes the condition $w_k \geq w_i$; otherwise, Equation (30) would prohibit the vertex k from sending a message such that $\nu_{k\to i} \geq w_i$. Given such w_k, the probability with which the vertex k with weight ranging from w_k to $w_k + \mathrm{d}w_k$ sends the message $\nu_{k\to i}$ ($\geq w_i$) is $\mathrm{d}w_k\ g(w_k) \int_{w_i^-}^{w_k} \mathrm{d}\nu_{k\to i}\, f(\nu_{k\to i}; w_k)$. Taking the integral over the weight distribution, we obtain

$$\begin{aligned} &P(0; w_i) = \int_{w_i^-}^{+\infty} \mathrm{d}w_k\, g(w_k) \int_{w_i^-}^{w_k} \mathrm{d}\nu_{k\to i}\, f(\nu_{k\to i}; w_k) \\ &= \int_{w_i^-}^{+\infty} \mathrm{d}w_k\, g(w_k) \int_{w_i^-}^{w_k} \mathrm{d}\nu_{k\to i}\, \delta(\nu_{k\to i})P(0; w_k) \\ &+ \int_{w_i^-}^{+\infty} \mathrm{d}w_k\, g(w_k) \int_{w_i^-}^{w_k} \mathrm{d}\nu_{k\to i}\, f_m(\nu_{k\to i}; w_k) \\ &+ \int_{w_i^-}^{+\infty} \mathrm{d}w_k\, g(w_k) \int_{w_i^-}^{w_k} \mathrm{d}\nu_{k\to i}\, \delta(\nu_{k\to i} - w_k)P(w_k; w_k). \end{aligned} \tag{33}$$

As for $P(w_i; w_i)$, we consider the first case of Equation (31), i.e., $\nu_{k\to i} = 0$. The probability with which vertex k vertex with weight ranging from w_k to $w_k + \mathrm{d}w_k$ sends the message 0 is $\mathrm{d}w_k\ g(w_k)P(0; w_k)$. Therefore, $P(w_i; w_i)$

is

$$P(w_i; w_i) = \int_{0^-}^{+\infty} \mathrm{d}w_k\, g(w_k) P(0; w_k). \quad (34)$$

The problem is then to solve the integral equations (Equations (32) to (34)) given a specific weight distribution $g(w)$ and under the normalization condition of $f(\nu_{i\to j}; w_i)$, i.e.,

$$\begin{aligned} &\int_{0^-}^{w_i} \mathrm{d}\nu_{i\to j}\, f(\nu_{i\to j}; w_i) \\ &= P(0; w_i) + \int_{0}^{w_i} \mathrm{d}\nu_{i\to j}\, f_m(\nu_{i\to j}; w_i) + P(w_i; w_i) \\ &= 1. \end{aligned} \quad (35)$$

Constant Positive Weights

We first consider constant positive weights, i.e., $\forall i, j \in V: w_i = w_j$. Without loss of generality, we assume that all weights equal 1. Then we have

$$g(w) = \delta(w - 1). \quad (36)$$

Plugging Equation (36) into Equations (32) to (34) leads to

$$f_m(\nu_{i\to j}; w_i) = f(w_i - \nu_{i\to j}; 1) \quad (37)$$

$$P(0; w_i) = \Theta(1 - w_i) \int_{w_i^-}^{1} \mathrm{d}\nu_{k\to i}\, f(\nu_{k\to i}; 1) \quad (38)$$

$$P(w_i; w_i) = P(0; 1). \quad (39)$$

Multiplying $g(w_i)\,\mathrm{d}w_i$ and integrating over $(0, +\infty)$ on both sides of these equations, we have

$$f_m(\nu_{i\to j}; 1) = f_m(1 - \nu_{i\to j}; 1) \quad (40)$$

$$P(0; 1) = P(1; 1). \quad (41)$$

Since all messages are initialized to 0, Equation (18) implies that each message can only be 0 or 1. Therefore, the solution to Equations (40) and (41) is

$$f(\nu_{i\to j}; 1) = \frac{1}{2}\left[\delta(\nu_{i\to j} - 1) + \delta(\nu_{i\to j} - 0)\right]. \quad (42)$$

This equation shows that each message has a probability of $\frac{1}{2}$ to be equal to 0 and 1, respectively. Therefore, the total weight w_t of all incoming messages to a vertex has the probability

$$P_{mt}(w_t) = \begin{cases} \frac{1}{4}, & w_t = 0 \\ \frac{1}{2}, & w_t = 1 \\ \frac{1}{4}, & w_t = 2. \end{cases} \quad (43)$$

Combined with the fact that all vertices have constant weight 1, we have the expected total weight of the MWVC being $N/2$, where N is the number of vertices. This result matches the expectation that a minimum of $N/2$ vertices are required to cover all edges in a loop.

Uniformly Distributed Weights

We now consider the case of uniformly distributed weights over the interval $[0, w_0]$, i.e.,

$$g(w) = \frac{1}{w_0}\Theta(w)\Theta(w_0 - w), \quad (44)$$

where $w_0 > 0$ is a parameter of this distribution. Substituting Equation (44) into Equations (32) to (34) results in, for $0 < w_i \le w_0$,

$$\begin{aligned} f_m(\nu_{i\to j}; w_i) = &\frac{1}{w_0}\int_{w_i-\nu_{i\to j}}^{w_0} \mathrm{d}w_k\, f_m(w_i - \nu_{i\to j}; w_k) \\ &+ \frac{1}{w_0} P(w_i; w_i) \quad \text{for} \quad 0 < \nu_{i\to j} < w_i \end{aligned} \quad (45)$$

$$P(0; w_i) = \frac{1}{w_0}\int_{w_i^-}^{w_0} \mathrm{d}w_k \int_{w_i^-}^{w_k} \mathrm{d}\nu_{k\to i}\, f(\nu_{k\to i}; w_k) \quad (46)$$

$$P(w_i; w_i) = \frac{1}{w_0}\int_{0^-}^{w_0} \mathrm{d}w_k\, P(0; w_k), \quad (47)$$

where we implicitly used $P(w_i; w_i) = P(w_i - \nu_{i\to j}; w_i - \nu_{i\to j})$ when deriving Equation (45), since according to Equation (47), $P(w_i; w_i)$ is a constant. To solve Equation (45), we start by recognizing that $f_m(\nu_{i\to j}; w_i)$ only depends on $w_i - \nu_{i\to j}$. Letting $y = w_i - \nu_{i\to j}$, $f_m(\nu_{i\to j}; w_i)$ is a function of y (denoted by $h(y)$). Equation (45) can be written as

$$\begin{aligned} h(y) &= \frac{1}{w_0}\int_{y}^{w_0} \mathrm{d}w_k\, h(w_k - y) + \frac{1}{w_0}P(w_i; w_i) \\ &= \frac{1}{w_0}\int_{0}^{w_0 - y} \mathrm{d}z\, h(z) + \frac{1}{w_0}P(w_i; w_i), \end{aligned} \quad (48)$$

where a change of variable, $z = w_k - y$, is made in the last line. Taking the derivative with respect to y and noting that $P(w_i; w_i)$ is independent of y (from Equation (47)), we obtain

$$h'(y) = -\frac{1}{w_0} h(w_0 - y), \quad (49)$$

which is a linear idempotent differential equation (Falbo 2003). Its solution is

$$h(y) = h_0\left(\cos\left(\frac{y}{w_0} - \frac{1}{2}\right) - \sin\left(\frac{y}{w_0} - \frac{1}{2}\right)\right), \quad (50)$$

where h_0 is a constant to be determined. By plugging the definition of y into Equation (50), we have the solution to Equation (45):

$$\begin{aligned} f_m(\nu_{i\to j}; w_i) = &h_0\left[\cos\left(\frac{w_i - \nu_{i\to j}}{w_0} - \frac{1}{2}\right)\right. \\ &\left. - \sin\left(\frac{w_i - \nu_{i\to j}}{w_0} - \frac{1}{2}\right)\right]. \end{aligned} \quad (51)$$

$P(w_i, w_i)$ can be found by plugging the solution to $h(y)$ (from Equation (50)) into Equation (48):

$$P(w_i; w_i) = h_0 w_0\left(\cos\left(\frac{1}{2}\right) - \sin\left(\frac{1}{2}\right)\right). \quad (52)$$

We can now solve Equation (46). Substituting $f(\nu_{k\to i}; w_k)$ using Equation (28), we expand Equation (46) as

$$\begin{aligned} P(0; w_i) = &\frac{1}{w_0}\int_{w_i}^{w_0} \mathrm{d}w_k \int_{w_i}^{w_k} \mathrm{d}\nu_{k\to i}\, f_m(\nu_{k\to i}; w_k) \\ &+ \frac{1}{w_0}\int_{w_i}^{w_0} \mathrm{d}w_k\, P(w_k; w_k). \end{aligned} \quad (53)$$

Plugging in Equations (51) and (52), the solution to Equation (53) is given by

$$P(0;w_i) = h_0 w_0 \Bigg[\cos\left(\frac{1}{2}\right) + \sin\left(\frac{1}{2}\right) - \sin\left(\frac{w_i}{w_0} - \frac{1}{2}\right) - \cos\left(\frac{w_i}{w_0} - \frac{1}{2}\right) \Bigg]. \tag{54}$$

In order to determine h_0, we use the normalization property of a probability distribution. Solving Equation (35) by substituting Equations (51), (52) and (54) fixes h_0 to

$$h_0 = \frac{1}{w_0\left(\cos\left(\frac{1}{2}\right) + \sin\left(\frac{1}{2}\right)\right)}. \tag{55}$$

Substituting Equation (55) into Equations (51), (52) and (54) leads to

$$f_m(v_{i\to j}; w_i) = \frac{1}{w_0}\left(\cos\left(\frac{w_i - \nu_{i\to j}}{w_0}\right) - \alpha \sin\left(\frac{w_i - \nu_{i\to j}}{w_0}\right)\right) \tag{56}$$

$$P(0;w_i) = 1 - \alpha\cos\left(\frac{w_i}{w_0}\right) - \sin\left(\frac{w_i}{w_0}\right) \tag{57}$$

$$P(w_i;w_i) = \alpha, \tag{58}$$

where $\alpha = \frac{1-\tan\left(\frac{1}{2}\right)}{1+\tan\left(\frac{1}{2}\right)} \approx 0.293$.

With $f(\nu_{i\to j}; w_i)$ expressed in closed form, we can calculate quantities such as the average weight contribution per vertex $\bar{w}$ to the total weight of an MWVC. In the case of a finite graph, $\bar{w}$ corresponds to the total weight of MWVC divided by the number of vertices. For a loop of infinite size, a given vertex of weight w_i is included in an MWVC iff $w_i < \nu_{j\to i} + \nu_{k\to i}$ (we ignore the case of $w_i = \nu_{j\to i} + \nu_{k\to i}$ since it has zero probability to occur), where vertices k and j are adjacent to vertex i vertex. Integrating over the weight distributions for w_j and w_k, and over the probability density for $\nu_{j\to i}$ and $\nu_{k\to i}$, we obtain

$$\bar{w} = \int_{0^-}^{+\infty} \mathrm{d}w_j\, g(w_j) \int_{0^-}^{+\infty} \mathrm{d}w_k\, g(w_k) \times \int_{0^-}^{w_j} \mathrm{d}\nu_{j\to i}\, f(\nu_{j\to i}; w_j) \int_{0^-}^{w_k} \mathrm{d}\nu_{k\to i}\, f(\nu_{k\to i}; w_k) \times \int_{0^-}^{\nu_{j\to i}+\nu_{k\to i}} \mathrm{d}w_i\, w_i g(w_i), \tag{59}$$

For the uniform distribution of the weights (Equations (56) to (58)), we have

$$\bar{w} = \frac{1+\sin(1) - 2\cos(1)}{2+2\sin(1)} w_0 \approx 0.2066 w_0. \tag{60}$$

Numerical Experiments

In this section, we verify our analysis of the MSMP algorithm for the MWVC problem on a loop. Since the MSMP

Algorithm 1: `LoopMWVC` finds the total weight of an MWVC of a loop.

Function `LoopMWVC` (W)
- **Input:** W: An array of weights of vertices in a loop.
- **Output:** The total weight of an MWVC of the loop.
- $N := |W|$;
- **if** $N \in \{0, 1\}$ **then return** 0;
- **if** $N = 2$ **then return** $\min\{W[0], W[1]\}$;
- **if** $N = 3$ **then return** $\sum_{w\in W} w - \max_{w\in W} w$;
- $w_1 := W[N-1] + \texttt{PathMWVC}(W[0:N-2])$;
- $w_2 := W[0] + W[N-2] + \texttt{PathMWVC}(W[1:N-3])$;
- **return** $\min\{w_1, w_2\}$;

Function `PathMWVC` (W)
- **Input:** W: An array of weights of vertices in a path.
- **Output:** The total weight of an MWVC of the path.
- $N := |W|$;
- *opt* := an empty array of length $N+1$;
- $opt[0] := 0, opt[1] := 0$;
- **for** $i \leftarrow 2$ **to** N **do**
  - $opt[i] := \min\{W[i-1]+opt[i-1], W[i-2]+opt[i-2]\}$;
- **return** $opt[N]$;

algorithm is known to produce optimal solutions for trees, our study of its behavior on loops is a fundamental step towards understanding the general case. We conducted the following experiment to verify our analysis on loops with uniformly distributed weights. The parameters used were the maximum weight w_0 and the loop length N. Without loss of generality, we fixed the value of w_0 to 1, so that according to Equation (60), the expected average weight contribution per vertex is $\bar{w} \approx 0.2066$ asymptotically. We varied N exponentially from 20 to 10^5 to generate 16 values of N within this range. For each value of N, we generated 50 loops of size N with uniformly distributed weights. For each loop, we computed the total weight of an MWVC using the MSMP algorithm and divided it by N to obtain $\bar{w}$. We also computed $\bar{w}$ of these loops using a simple linear-time dynamic programming-based approach (Algorithm 1).

Figure 2 shows the results of the numerical experiments. As the size of the loop increases, the actual average sizes of the MWVCs become closer to those predicted by the analytical results. This observation seems to demonstrate that our analytical framework works well, at least on loops. In our experiments (as well as in the analytical solution in section "Constant Positive Weights"), we also observed that, upon convergence, the MSMP algorithm always produces optimal solutions asymptotically. This seems to support the conjecture that, the MSMP algorithm for the MWVC problem on general loopy graphs may be effective, as long as it converges. Indeed, it has been shown that it is beneficial to first convert a WCSP instance to an MWVC problem instance using the concept of the constraint composite graph (Xu, Kumar, and Koenig 2017). In particular, the effectiveness of the MSMP algorithm can be significantly improved on

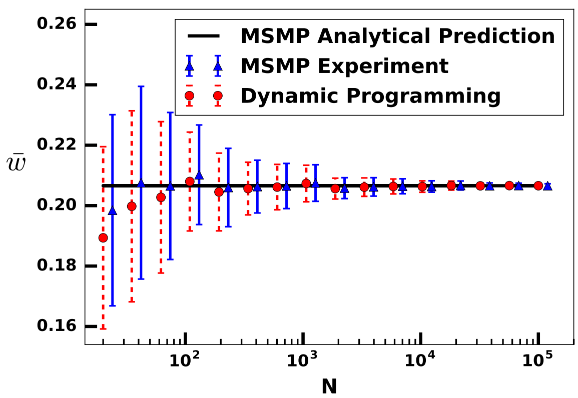

Figure 2: Shows the average weight contribution per vertex $\bar{w}$ of an MWVC for loops of different sizes and with weights uniformly distributed over the interval $[0, 1]$. The black solid line indicates the asymptotic value of $\bar{w}$ as N tends to infinity, predicted by the analytical study (Equation (60)). The blue triangles represent the values of $\bar{w}$ estimated by the MSMP algorithm, and the blue solid vertical bars indicate the standard deviation over 50 problem instances of the same size. The red circles indicate the true values of $\bar{w}$ computed by dynamic programming (Algorithm 1), and the red dashed vertical bars show the standard deviation over 50 problem instances of the same size. For visual clarity, the horizontal positions of the blue triangles are slightly shifted to the right of the corresponding red circles in the plot.

the MWVC problem reformulation of the WCSP; and (Xu, Kumar, and Koenig 2017) demonstrates this effectiveness empirically. In this paper, we support the same general strategy of first reformulating a given combinatorial optimization problem as the MWVC problem; but we do this by creating a strong analytical framework for understanding the MSMP algorithm.

Conclusions and Future Work

In this paper, we developed the MSMP algorithm for the MWVC problem and studied its effectiveness. We showed that this algorithm generalizes the WP algorithm known for the MVC problem. While the MSMP algorithm is analytically well studied on trees, we took the first fundamental step to build a systematic analytical framework towards understanding its behavior on general graphs. We analytically derived the total weight of an MWVC of infinite loops with constant and uniformly distributed weights on vertices. We showed that in both cases, our analytical results matched those of theoretical expectations and experiments, respectively. Our contributions support the general strategy of using the MSMP algorithm on the MWVC problem reformulation of a given combinatorial optimization problem (instead of directly on it). In particular, we created a strong analytical framework for understanding the MSMP algorithm on the MWVC problem and consequently on all combinatorial optimization problems that can be reformulated as the MWVC problem.

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

Bistarelli, S.; Montanari, U.; Rossi, F.; Schiex, T.; Verfaillie, G.; and Fargier, H. 1999. Semiring-based CSPs and valued CSPs: Frameworks, properties, and comparison. *Constraints* 4(3):199–240.

Cai, S.; Su, K.; Luo, C.; and Sattar, A. 2013. NuMVC: An efficient local search algorithm for minimum vertex cover. *Journal of Artificial Intelligence Research* 46(1):687–716.

Chen, J.; Kanj, I. A.; and Xia, G. 2006. Improved parameterized upper bounds for vertex cover. In *the International Symposium on Mathematical Foundations of Computer Science*, 238–249.

Cormen, T. H.; Leiserson, C. E.; Rivest, R. L.; and Stein, C. 2009. *Introduction to Algorithms (3rd Edition)*. MIT Press.

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

Falbo, C. E. 2003. Idempotent differential equations. *Journal of Interdisciplinary Mathematics* 6(3):279–289.

Johnson, D. J., and Trick, M. A., eds. 1996. *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge*. American Mathematical Society.

Koller, D., and Friedman, N. 2009. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

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*.

Mézard, M., and Montanari, A. 2009. *Information, Physics, and Computation*. Oxford University Press.

Mézard, M., and Zecchina, R. 2002. Random k-satisfiability problem: From an analytic solution to an efficient algorithm. *Physical Review E* 66(5):056126.

Niskanen, S., and Östergård, P. R. J. 2003. Cliquer user's guide, version 1.0. Technical Report T48, Communications Laboratory, Helsinki University of Technology, Espoo, Finland.

Richter, S.; Helmert, M.; and Gretton, C. 2007. A stochastic local search approach to vertex cover. In *Proceedings* *of the Annual German Conference on Artificial Intelligence (Künstliche Intelligenz)*, 412–426.

Sandholm, T. 2002. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence* 135(1):1–54.

Weigt, M., and Zhou, H. 2006. Message passing for vertex covers. *Physical Review E* 74(4):046110.

Xu, H.; Kumar, T. K. S.; and Koenig, S. 2016. A new solver for the minimum weighted vertex cover problem. In *the International Conference on Integration of Artificial Intelligence and Operations Research Techniques in Constraint Programming*, 392–405.

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.

Yedidia, J. S.; Freeman, W. T.; and Weiss, Y. 2003. Understanding belief propagation and its generalizations. *Exploring Artificial Intelligence in the New Millennium* 8:239–269.