Improved Algorithms for Vertex Cover with Hard Capacities on Multigraphs and Hypergraphs

Wang Chi Cheung*, Michel X. Goemans†, Sam Chiwai Wong‡

Abstract
In this paper, we consider the minimum unweighted Vertex Cover problem with Hard Capacity constraints (VCHC) on multigraphs and hypergraphs. Given a graph, the objective of VCHC is to find a smallest multiset of vertices that cover all edges, under the constraints that each vertex can only cover a limited number of incident edges, and the number of available copies of each vertex is bounded. This problem generalizes the classical unweighted vertex cover problem. Here we restrict our attention to unweighted instances, since the weighted version of VCHC is as hard as the set cover problem, as shown by Chuzhoy and Naor (FOCS 2002).

We obtain improved approximation algorithms for VCHC on multigraphs and hypergraphs. This problem has first been studied by Saha and Khuller (ICALP 2012). They proposed a 38-approximation for multigraphs, and a max \(65f,65\) approximation for hypergraphs, where \(f\) is the size of the largest hyperedge. In this paper, we significantly improve these approximation ratios to \(1 + 2 + \sqrt{\frac{2}{3}} < 2.155\) and \(2f\) respectively. In the case of multigraphs, our approximation ratio is very close to the longstanding bound of 2 for the classical vertex cover problem. Our algorithms consist of a two-step process, each based on rounding an appropriate linear program. In particular, for multigraphs, the analysis in the second step relies on identifying a matching structure within any extreme point solution.

Furthermore, we consider the partial VCHC problem in which one only needs to cover all but \(f\) edges. We propose a generic reduction from partial VCHC on \(f\)-hypergraphs to VCHC on \((f+1)\)-hypergraphs, with a small loss in the approximation factor. In particular, we present a \((2f + 2)(1 + \epsilon)\)-approximation algorithm for partial VCHC on \(f\)-hypergraphs.

1 Introduction
The minimum vertex cover problem is one of the most well-studied combinatorial optimization problems. In this classical problem, we are given a graph \(G = (V,E)\) and a weight for every vertex \(v \in V\), and the objective is to select a minimum weight subset \(U \subseteq V\) such that for all \(e \in E\), \(e\) is incident to at least one vertex \(v \in U\). It is well known that this problem has a 2-approximation algorithm by linear programming techniques (Hochbaum[8] and Bar-Yehuda and Even[1]).

This also extends to an \(f\)-approximation algorithm for \(f\)-hypergraphs, which are hypergraphs with largest edge size \(\leq f\). On the other hand, assuming the Unique Game Conjecture, [9] shows that, for any \(f \geq 2\), approximating the vertex cover problem on \(f\)-hypergraphs better than \(f - \epsilon\) is \(\text{NP}\)-hard, for any \(\epsilon > 0\).

In this paper, we offer improved approximation algorithms for the unweighted minimum vertex cover with hard capacity constraints (VCHC) on multigraphs and hypergraphs. In these problems, we are given a graph \(G = (V,E)\), which can have parallel edges. The objective is to find a minimum vertex cover, but under the constraints that each vertex \(v\) can cover at most \(k_v\) incident edges and that we can only select up to \(m_v\) copies of \(v\) for each vertex \(v\). These problems generalize the classical unweighted vertex cover problem.

Throughout this paper, we focus on unweighted instances, i.e., all vertices have weight 1. It is because the weighted version of VCHC for simple graphs is already as hard as set cover (Chuzhoy and Naor[4]), while the unweighted versions for multigraphs and hypergraphs admit constant approximation algorithm (Saha and Khuller[10]). This property makes many standard techniques such as primal-dual algorithms, LP rounding or iterative rounding methods not directly applicable for a constant approximation, since very often those techniques provide the same approximation ratio on weighted and unweighted instances.

1.1 Prior Work. The work on minimum capacitated vertex cover problems falls in two categories: hard capacities, where there is an upper bound \(m_v\) on the number of available copies of \(v\), and soft capacities, where there is no upper bound, i.e., \(m_v = \infty\). The weighted vertex cover problem with soft capacity constraints is first proposed by Guha et al.[7], which give a 2-approximation primal-dual algorithm. Subsequently, another 2-approximation via dependent randomized rounding is provided in Gandhi et al.[6].

On the other hand, the vertex cover problem with hard capacity constraints (VCHC) on simple graphs is first studied by Chuzhoy and Naor [4]. They propose an elegant 3-approximation algorithm for the unweighted
problem, through randomized rounding followed by a patching procedure. On the other hand, it is shown in [4] that the weighted version of VCHC is as hard as the set cover problem. Thus, subsequent work on VCHC has focused on unweighted instances. In particular, a 2-approximation algorithm for VCHC on simple graphs has been obtained by Gandhi et al. [5], by refining the approach in [4].

Nevertheless, as pointed out by Saha and Khuller [10], the algorithms in [4, 5] have no approximation guarantee for multigraphs. These are indeed randomized algorithms, and the presence of parallel edges induces positive correlation for some random variables in their analyses. This hinders the use of certain concentration inequalities vital for obtaining a constant approximation guarantee. In fact, achieving a constant approximation guarantee for VCHC for multigraphs is an open problem posed in [4]. Recently, this was settled by Saha and Khuller [10], who derive a 34-approximation algorithm in the case of unit multiplicities ($m_v = 1$ for all $v \in V$), and a 38-approximation for general multiplicities. They also give a max{6$f$, 65}-approximation algorithm for $f$-hypergraphs. These algorithms are based on a rounding paradigm, followed by a randomized clustering procedure on an appropriate multiset-multicover instance.

The partial vertex cover problem is a generalization of the vertex cover problem in which we are given an integer $\ell$ and we only need to cover all but $\ell$ edges in the graph. Capacitated versions are similarly generalized. The partial vertex cover problem without capacity constraints was first studied by Bshouty and Burroughs [3], which give a 2-approximation algorithm. For partial capacitated vertex cover with soft capacities, Bar-Yehuda et al. [2] derive a 2-approximation algorithm for simple graphs and a 3-approximation algorithm for multigraphs. Both algorithms work for weighted instances and are based on local ratio techniques. For the unweighted partial vertex cover with hard capacities, Saha and Khuller [10] have announced without details an $O(f)$-approximation algorithm for $f$-hypergraphs.

### 1.2 Our Contributions and Approach

Our main contributions are the following. We obtain:

- A 1 + 2/\sqrt{3} < 2.155-approximation algorithm for VCHC on multigraphs, which improves over the previous approximation ratio of 38 in [10]. In particular, our ratio is very close to the longstanding bound of 2 for classical vertex cover problem.

- A 2$f$-approximation for VCHC on hypergraphs, which improves over the previous approximation ratio of max{65, 6$f$} in [10].

- A generic reduction from partial VCHC on $f$-hypergraphs to VCHC on $(f + 1)$-hypergraphs. In particular, we obtain a $(2f + 2)(1 + \epsilon)$-approximation for partial VCHC on $f$-hypergraphs.

Our algorithm for multigraphs is a two-step process, each based on rounding an appropriate linear program. In the first step, we solve a natural LP relaxation to VCHC on multigraphs. Based on an optimal solution to this LP, we select a multiset $U$ of vertices such that every edge has one of its ends in $U$; viewed as a set, $U$ is a vertex cover. However, $U$ might not have enough capacity to cover all edges. Thus, in the second step, for each uncovered edge, we associate its deficit with its end in $U$. We then construct a covering LP whose solution provides enough coverage to offset the deficits of all vertices $u \in U$. By rounding up an extreme point solution to this covering LP and adding it to $U$, we have now enough capacity to cover all the edges.

We would like to contrast our algorithm with previous algorithms on VCHC. On one hand, the algorithms in [4, 5, 10] are also two-step processes, similar to ours. On the other hand, they perform randomized rounding in their second steps, while we solve an optimization problem. Therefore, unlike [4, 5], our approach allows us to bypass the issue of positive correlation among parallel edges. Furthermore, our optimization approach allows us to carry out a tighter analysis than that in [10], which involves a complicated analysis on its randomized clustering procedure. This leads to a significant improvement in the approximation guarantee. Our analysis in the second step relies on identifying a matching structure within any extreme point solution of the covering LP, which is vital for establishing the approximation ratio.

Our algorithm for hypergraphs is also a two-step process. Similar to the case of multigraphs, we first solve a natural LP relaxation of VCHC on hypergraphs for an optimal solution, and define a multiset of vertices $U$ such that every hyperedge has some vertex in $U$. However, in the second step, the construction of the covering LP requires more work. Indeed, an uncovered edge may have more than one vertex in $U$, and it is unclear which $u \in e \cap U$ to associate the deficit of the hyperedge $e$ with. We resolve this problem by splitting the deficit of $e$ among $e \cap U$ appropriately. We then set up the covering LP in a similar way to the case of multigraphs, and round up its extreme point solution and include the corresponding vertices into the solution.

**Organization.** In Section 2, we define the minimum Vertex Cover with Hard Capacities problem (VCHC) formally. In Section 3, we present a family of approximation algorithms for VCHC on multigraphs based on a rounding threshold, and this high-
lights the main ideas in the analysis. Section 4 provides an improved \((1 + 2/\sqrt{3})\)-approximation algorithm for VCHC on multigraphs by randomizing the threshold. In Section 5, we obtain a \(2f\)-approximation algorithm to VCHC on \(f\)-hypergraphs. In Section 6, we demonstrate a generic reduction from partial VCHC to VCHC. Concluding remarks are in Section 7.

2 Problem Statement

We start by specifying some notation for multigraphs. A multigraph \(G = (V, E)\) may contain several edges between two vertices \(u\) and \(v\), and they will all be denoted by \(uv\) even though we can uniquely identify them as different members \(e_i\)'s in \(E\). For an edge \(e = uv\) between \(u\) and \(v\), we often write that \(u \in e\) (and similarly that \(v \in e\)).

The minimum Vertex Cover problem with Hard Capacity constraints (VCHC) is formally defined as follows. An instance of VCHC is specified by \((V, E, k, m)\), where

- \(G = (V, E)\) is the input multigraph,
- For each \(v \in V\), \(m_v\) denotes the maximum number of copies of \(v\) one can select,
- For each \(v \in V\), \(k_v\) is the number of incident edges (a copy of) \(v\) can cover.

A solution to VCHC consists of \((x, y) = ((x_v)_{v \in V}, (y(e, v))_{e \in E, v \in e})\). Here, \(x_v\) is the number of copies of vertex \(v\) selected, and the assignment variable \(y(e, v) \in \{0, 1\}\) represents whether edge \(e\) is covered by \(v\), for each \(e \in E\) and \(v \in e\). A solution \((x, y)\) is feasible for VCHC iff

1. For all \(v \in V\): \(x_v \in \{0, 1, \cdots, m_v\}\),
2. For all \(e = uv \in E\): \(y(e, u) + y(e, v) = 1\) (i.e. any edge must be covered by one of its endpoints),
3. For all \(v \in V\): \(|\{e : y(e, v) = 1\}| \leq k_v x_v\) (i.e. the total number of edges assigned to \(v\) does not exceed its total capacity).

The objective of VCHC is to find a feasible solution \((x, y)\) for VCHC that minimizes \(\sum_{v \in V} x_v\), the size of the vertex cover. We note that VCHC generalizes the classical minimum vertex cover problem, which is already \(\text{NP}\)-hard. In light of this, we are providing efficient algorithms able to find approximate solutions to VCHC. Our approach is based on rounding a fractional solution to the following LP relaxation, which has been used extensively in the literature [4, 5, 10].

\[
\begin{align*}
\text{min} & \quad \sum_{v \in V} x_v \\
\text{s.t.} & \quad y(e, u) + y(e, v) = 1 & \forall e = uv \in E \\
& \quad y(e, u) \leq x_u & \forall e = uv \in E, u \in e \\
& \quad \sum_{e \in \delta(u)} y(e, v) \leq k_u x_u & \forall u \in V \\
& \quad x_u \leq m_u & \forall u \in V \\
& \quad x, y \geq 0
\end{align*}
\]

The following lemma shows that, when constructing a feasible solution to VCHC, we only need the integrality of \(x\), and not of \(y\), as was established in Chuzhoy and Naor [4]. This follows easily by the integrality of flows in networks with integer capacities. We refer the reader to [4] for a proof, or to the forthcoming Lemma 5.1 for a more general statement.

**Lemma 2.1. (Chuzhoy and Naor [4])** If \((x, y)\) is feasible for LP 2.1, and \(x\) is integral, there exists an integral \(y'\) such that \((x, y')\) is feasible for LP 2.1, and \(y'\) can be found efficiently by a maximum flow computation.

3 A \(3\)-Approximation to VCHC on Multigraphs

In this section, we present and analyze a \((1 + 1/\alpha)\)-approximation algorithm \(\text{ALG}_{\alpha}\) for VCHC on multigraphs, where \(\alpha \in [0, 1/2]\). Choosing \(\alpha = 1/2\) will then yield a \(3\)-approximation. On the other hand, the purpose of presenting \(\text{ALG}_{\alpha}\) for \(\alpha \in [0, 1/2]\) is to offer insight into the two-step process of the algorithm, and to highlight the key ideas in its analysis. This also serves as a warm-up for the improved algorithm in the next section, where \(\alpha\) is selected randomly.

3.1 The \((1 + \frac{1}{\alpha})\)-Approximation Algorithm. The high level idea of \(\text{ALG}_{\alpha}\), where \(\alpha \in [0, 1/2]\), is as follows. We first solve LP 2.1 for an optimal solution \((x^*, y^*)\), and round up all vertices \(v\) with \(x^*_v \geq \alpha\). Since those vertices might not have enough capacity to cover all edges\(^1\), we select the remaining vertices by considering another LP, namely LP 3.2. Finally, we assign the coverage of the edges to the selected vertices by Lemma 2.1.

We now describe our approximation algorithm \(\text{ALG}_{\alpha}\). In anticipation of the next section, we introduce another algorithm \(\text{VC}(\hat{x}, y, \alpha)\), which is a subroutine in \(\text{ALG}_{\alpha}\) and the approximation algorithm in Section 4.

\(^1\)as opposed to the situation in the classical vertex cover problem without capacities.
\(\delta'(u)\)

\[E' \quad \text{u} \quad \delta'(u)\]

\[\text{W} \quad \text{U} \quad \text{Z}\]

Figure 1: ALG\(_\alpha\) partitions the vertex set \(V\) into \(W, U, Z\). Note that there is no edge inside \(W \cup Z\).

\(\text{ALG}_\alpha\) is formally stated as follows:

1. Solve LP 2.1 for an optimal solution \((x^*, y^*)\).

2. Run the algorithm VC\((x^*, y^*, \alpha)\), which returns a solution \((x', y')\) feasible for VCHC.

Next, VC\((\hat{x}, \hat{y}, \alpha)\) is an algorithm that takes as input a solution \((\hat{x}, \hat{y})\) feasible for LP 2.1 and a threshold \(\alpha \in [0, 1/2]\). It outputs an \((x', y')\) feasible for VCHC.

VC\((\hat{x}, \hat{y}, \alpha)\) is formally stated as follows:

1. Given the input \((\hat{x}, \hat{y}, \alpha)\), partition \(V\) into \(U \cup W \cup Z\) where, as illustrated in Fig 1:
   - \(U = \{u \in V : \hat{x}_u \geq \alpha\}\),
   - \(W = \{w \in V : 0 < \hat{x}_w < \alpha\}\),
   - \(Z = \{z \in V : \hat{x}_z = 0\}\).

Note that there is no edge in \(E\) within \(W \cup Z\). Indeed, such an edge \(e = uw\) with \(u, v \in W \cup Z\) would imply

\[1 = \hat{y}(e, u) + \hat{y}(e, v) \leq \hat{x}_u + \hat{x}_v < 2\alpha \leq 1,\]

which is a contradiction.

2. Define the partial solution \(x'\), where

\[x'_u = \begin{cases} \lceil \hat{x}_u \rceil & \text{if } u \in U \\ 0 & \text{if } u \in Z, \end{cases}\]

and note that we have not defined \(x'_w\) for \(w \in W\) yet.

3. The capacities provided by \(x'_u\) for \(u \in U\) are enough to cover the edges entirely within \(U\) and also the edges between \(U\) and \(Z\), but may not be enough to fully cover the edges in \(E' := \{e = uw \in E : u \in U, w \in W\}\). For this reason, we design another linear program (LP 3.2) defined on \(W\) whose solutions will provide us with enough capacity to cover all the edges of the multigraph. This statement is motivated in the construction below and will be proved later in the section.

For edge \(e = uw\) with \(u \in U\) and \(w \in W\), our partial solution \(x'\) allows \(u\) to cover \(\hat{y}(e, u)\), but there might be a deficit of \(\hat{y}(e, w)\) for this edge which we can associate to \(u\). The total deficit associated to \(u \in U\) is

\[r(u) = \sum_{w \in W : e = uw \in E} \hat{y}(e, w) = \sum_{e \in \delta'(u)} \hat{y}(e, w),\]

using the notation \(\delta'(u) = \delta(u) \cap E'\). On the other hand, if we do select a vertex \(w \in W\) to be part of our vertex cover, its value \(\hat{x}_w\) is scaled up to 1 and therefore its coverage to \(e = uw\) can be increased from \(\hat{y}(e, w)\) to \(\hat{y}(e, w)/\hat{x}_w\). Vertex \(w\), if selected, can provide

\[M(u, w) = \sum_{e = uw \in E'} \frac{\hat{y}(e, w)}{\hat{x}_w}\]

units of coverage to \(u \in U\), where we are summing over all edges between \(u\) and \(w\) in our multigraph.

This motivates the following covering linear program:

\[\min \sum_{w \in W} z_w\]

(3.2a) \[\text{s.t. } \sum_{w \in W} M(u, w) z_w \geq r(u) \quad \forall u \in U\]

(3.2b) \[0 \leq z_w \leq 1 \quad \forall w \in W.\]

In this step, we compute an optimal extreme point solution \(z^*\) to this linear program.

4. Define \(x'_w = \lceil z^*_w \rceil\) for all \(w \in W\). Note that \(x'\) is now integral.

5. Compute an integral assignment \(y'\) via network flows as in Lemma 2.1, and output \((x', y')\).

Before the analysis, we would like to compare our algorithm with the previous algorithms on VCHC [4, 5, 10]. The first step of both our algorithm and the previous ones is based on LP rounding. The difference is in the second step, where essentially all previous algorithms obtain a feasible instance from \(x^*\) (the optimal solution to 2.1) by randomized rounding and patching (see e.g. [4, 5]). Our insight is that, instead of using \(x^*\), writing down the actual requirements explicitly as in LP 3.2 produces a sparse extreme
point solution which can be easily converted to a feasible solution. Our approach is deterministic and the structure of the extreme point solution enables a more elegant analysis.

3.2 Feasibility of \( \text{ALG}_\alpha \). Before establishing the approximation guarantee, we first prove that the output \((x', y')\) of \( \text{ALG}_\alpha \) is indeed a feasible VCHC in the following theorem:

**Theorem 3.1.** Let \((\hat{x}, \hat{y})\) be feasible for LP 2.1, and \(0 \leq \alpha \leq \frac{1}{2}\). Consider any feasible solution \(z\) to the corresponding LP 3.2 in \(VC(\hat{x}, \hat{y}, \alpha)\). Then for

\[
x_v = \begin{cases} 
[\hat{x}_v] & \text{if } v \in U, \\
\hat{z}_v & \text{if } v \in W, \\
0 & \text{if } v \in Z,
\end{cases}
\]

there exists \(y\) such that \((x, y)\) is feasible for LP 2.1.

In the following proof, we explicitly define a \(y\) such that \((x, y)\) is feasible. While we need to check for all the constraints (2.1a)-(2.1e), the main part of the proof is to show that \(x\) has enough capacity to cover all the edges in \(E'\). This justifies the construction of LP 3.2 in \(VC(\hat{x}, \hat{y}, \alpha)\).

**Proof.** In this proof, we define a \(y\) such that \((x, y)\) is feasible:

- For \(e = uw \in E'\), where \(u \in U\), \(w \in W\), define
  \[
y(e, w) = \frac{\hat{y}(e, w)}{\hat{x}_w} z_w, \quad y(e, u) = 1 - y(e, w).
\]

- For \(e = uw \in E \setminus E'\), define
  \[
y(e, w) = \hat{y}(e, w), \quad y(e, v) = \hat{y}(e, v).
\]

We claim that \((x, y)\) is feasible for LP 2.1. First, for all \(e \in E, v \in e\), we have \(0 \leq y(e, v) \leq 1\), and this implies that (2.1a), (2.1e) are satisfied. We show that (2.1b) is also satisfied for each vertex:

- When \(u \in U\) and \(e \in \delta(u)\), we have
  \[
y(e, u) \leq 1 = [\hat{x}_u] = x_u,
\]

- When \(w \in W\) and \(e \in \delta(w)\), we have
  \[
y(e, w) = \frac{\hat{y}(e, w)}{\hat{x}_w} z_w \leq z_w = x_w,
\]

- When \(z \in Z\) and \(e \in \delta(z)\), we have \(y(e, z) = 0 = x_z\).

Moreover, (2.1d) is satisfied for all vertices:

- For \(u \in U\), we have
  \[
x_u = [\hat{x}_u] \leq m_u,
\]
as \(\hat{x}_u \leq m_u\) and \(m_u\) is integral.

- For \(w \in W\), by the definition of \(W\), we have \(\hat{x}_w > 0\). It implies that \(m_w \geq 1\), thus \(x_w = z_w \leq 1 \leq m_w\).

- For \(z \in Z\), we have \(x_z = 0 \leq m_z\).

Finally, it remains to check (2.1c). For \(w \in W\), we have

\[
\sum_{e \in \delta(w)} y(e, w) = \sum_{e \in \delta(w)} \frac{\hat{y}(e, w)}{\hat{x}_w} z_w \leq \frac{k_w \hat{x}_w}{\hat{x}_w} z_w = k_w z_w = k_w x_w.
\]

For \(u \in U\), we have

\[
\sum_{e \in \delta(u)} y(e, u) = \sum_{e \in \delta(u)} \hat{y}(e, u) + \sum_{e = uw \in \delta'(u)} \left(1 - \frac{\hat{y}(e, w)}{\hat{x}_w} z_w\right)
\]

\[
= \sum_{e \in \delta(u) \setminus \delta'(u)} \hat{y}(e, u) + |\delta'(u)| - \sum_{w \in W} M(u, w) z_w
\]

\[
\leq \sum_{e \in \delta(u) \setminus \delta'(u)} \hat{y}(e, u) + |\delta'(u)| - r(u)
\]

\[
= \sum_{e \in \delta(u) \setminus \delta'(u)} \hat{y}(e, u) + \sum_{e \in \delta'(u)} \hat{y}(e, u) \leq k_u \hat{x}_u \leq k_u x_u.
\]

Finally, for \(z \in Z\), \(\sum_{e \in \delta(z)} y(e, z) = 0 = x_z\). Altogether, \((x, y)\) is feasible to LP 2.1.

As we set \(x' = [z'_{w}] \geq z'_{w}\) for \(w \in W\), Theorem 3.1 then implies that the integral solution \(x'\) is a feasible capacitated vertex cover. We emphasize that we do not need the optimality of \((\hat{x}, \hat{y})\) for Theorem 3.1 to hold, and that LP 3.2 depends on the solution \((\hat{x}, \hat{y})\) as well as \(\alpha\); this will be important in the next section. Finally, Theorem 5.1 will generalize Theorem 3.1.

3.3 Approximation Guarantee for \(\text{ALG}_\alpha\). In this section, we establish the approximation guarantee of our proposed algorithm. First, the cost \(C\) of the output solution \((x', y')\) is bounded as follows:

\[
C = \sum_{v \in V} x'_v
\]

\[
= \sum_{u \in U} [x'_u] + \sum_{w \in W} [z'_{w}] 
\]

\[
\leq \sum_{u \in U} [x'_u] + \sum_{w \in W} z'_{w} + |W|,
\]

(3.3)
where \( W_f := \{ w \in W : 0 < z_w^* < 1 \} \) is the set of vertices in \( W \) with fractional value \( z^* \). Now we proceed to bounding the second term \( \sum_{w \in W} z_w^* \) and third term \( |W_f| \).

First, we show the following bound on the second term \( \sum_{w \in W} z_w^* \), by our choice of LP 3.2.

**Lemma 3.1.** In \( VC(\hat{x}, \hat{y}, \alpha) \), where \((\hat{x}, \hat{y})\) is a feasible solution to LP 2.1 and \( 0 \leq \alpha \leq 1/2 \), the solution \( \{ \hat{x}_w \}_{w \in W} \) is feasible for LP 3.2. Therefore, \( \sum_{w \in W} z_w^* \leq \sum_{w \in W} \hat{x}_w \).

**Proof.** By definition of \( W \), we know that \( 0 \leq \hat{x}_w < \alpha < 1 \) for all \( w \in W \), so (3.2b) is satisfied. For (3.2a), we have for all \( u \in U \)

\[
\sum_{w \in W} M(u, w) \hat{x}_w = \sum_{w \in W} \sum_{e \subseteq uw} \frac{\hat{y}(e, w)}{\hat{x}_w} \hat{x}_w = \sum_{e \subseteq U} \hat{y}(e, w) = r(u).
\]

So \( \{ \hat{x}_w \}_{w \in W} \) is feasible for LP 3.2. \( \square \)

Next, we bound \( |W_f| \) by exploiting the sparsity of an extreme point of LP 3.2. In fact, we establish the existence of a matching between \( W_f \) and \( U \) that covers all vertices in \( W_f \). This structural result allows us to charge the cost of rounding up \( z_w^* \) partly to the vertices in \( U \). The charging here crucially uses the fact that the objective function is unweighted.

**Theorem 3.2.** For any extreme point solution \( z^* \) to 3.2, let \( W_f = \{ w \in W : 0 < z_w^* < 1 \} \) as above. Then there exists a matching between \( W_f \) and \( U \) that fully matches \( W_f \). Furthermore, we have \( |W_f| \leq \sum_{w \in V} \hat{x}_w \).

**Proof.** Since \( z^* \) is an extreme point solution of LP 3.2, there exists a set of \( W \) linearly independent inequalities in LP 3.2 that are satisfied by \( z^* \) with equalities. By renaming the vertices in \( U \) and \( W \), we have a system of \( |W| \) linearly independent equalities in \( |W| \) variables (recall \( z^* \in [0, 1]^{|W|} \)) of the following form:

\[
\begin{pmatrix}
M(u_1, w_1) & \cdots & M(u_1, w_t) \\
\vdots & \ddots & \vdots \\
M(u_t, w_1) & \cdots & M(u_t, w_t)
\end{pmatrix}
\begin{pmatrix}
z^*_1 \\
z^*_t
\end{pmatrix} =
\begin{pmatrix}
r(u_1) \\
r(u_t)
\end{pmatrix}.
\]

Here, the first \( t \) equalities are from the constraints (3.2a) and correspond to \( \{ u_1, \cdots, u_t \} \subseteq U \), and the last \( |W| - t \) equalities are from the constraints (3.2b). Let’s call the upper left matrix \( A \), and let \( W' = \{ w_1, \cdots, w_t \} \subseteq W \) be the set of \( w \in W \) that corresponds to the first \( t \) columns in the above set of equalities. Observe that \( W_f \) is a subset of \( W' \) since \( z_w^* \in \{ 0, 1 \} \) for \( w \notin W' \).

To exhibit the matching, notice that \( \text{det}(A) \neq 0 \) as there exists a permutation \( \sigma \) on \( \{ 1, 2, \cdots, t \} \) such that \( \prod_{i=1}^t M(u_i, w_{\sigma(i)}) \neq 0 \). Furthermore, if \( M(u, w) \neq 0 \), then \( uw \) must be an edge. Therefore, there is a matching between \( W' \) and \( U \) that fully matches \( W' \), and by restricting to the edges incident to \( W_f \), we get the desired matching between \( W_f \) and \( U \).

Finally, to bound \( |W_f| \), we have the following:

\[
|W_f| \leq |W'| \leq \sum_{i=1}^t (\hat{x}_{u_i} + \hat{x}_{w_i}) \leq \sum_{w \in V} \hat{x}_w. \quad \square
\]

Combining all the pieces, we establish the approximation guarantee of \( \text{ALG}_\alpha \).

**Theorem 3.3.** \( \text{ALG}_\alpha \) is a \( (1 + \frac{1}{\alpha}) \)-approximation algorithm for the VCHC problem on multigraphs.

**Proof.** For \( \text{ALG}_\alpha \), we have that \( (\hat{x}, \hat{y}) = (x^*, y^*) \). By (3.3), we have that the cost of the solution returned satisfies

\[
\sum_{v \in V} x_v^* \leq \sum_{u \in U} \sum_{w \in W} x_u^* + \sum_{v \in V} x_v^* \
\leq \left( \frac{1}{\alpha} + 1 \right) \sum_{u \in U} x_u^* + 2 \sum_{w \in W} x_w^* \leq \left( \frac{1}{\alpha} + 1 \right) \sum_{v \in V} x_v^*,
\]

where the first inequality follows by Lemma 3.1 and Theorem 3.2, the second one from the fact that \( x_u^* \geq \alpha \) for all \( u \in U \), and the third one by the fact that \( 0 \leq \alpha \leq \frac{1}{2} \). \( \square \)

Finally, choosing \( \alpha = 0.5 \) gives the desired 3-approximation algorithm.

**4 A \( (1 + 2/\sqrt{3}) \)-approximation to VCHC on Multigraphs via Random Threshold**

In the previous section, we establish that the approximation algorithm \( \text{ALG}_\alpha \) has a worst-case approximation ratio of \( 1 + 1/\alpha \) for \( \alpha \in [0, 1/2] \). In particular, the smallest worst-case ratio is 3 by choosing \( \alpha = 1/2 \). However, a given instance need not attain the worst-case ratio for every value of \( \alpha \). This suggests trying every possible value \( \alpha \) and outputting the best solution produced. However, for the sake of the analysis, it is easier to consider a randomized algorithm where the rounding threshold \( \alpha \) is chosen according to a probability distribution \( \mathcal{A} \). In this section, we show that for a suitable such distribution and a slight modification of the algorithm \( \text{ALG}_\alpha \), the corresponding randomized approximation algorithm \( \text{ALG}'_\alpha \) has a performance guarantee.

1719 Copyright © 2014.

by the Society for Industrial and Applied Mathematics.
of $1 + 2/\sqrt{3} < 2.155$, which improves upon the factor 3
from the previous section. Furthermore, the algorithm can be easily derandomized by trying all relevant values of $\alpha$.

We first describe informally our modification of the algorithm for a given value of $\alpha$. Instead of applying $\text{VC}(x^*, y^*, \alpha)$ as in the previous section, we apply $\text{VC}$ on a modified LP solution $(\hat{x}, \hat{y})$ obtained by first rounding up the $x_u^*$ for $u \in U$ and also assigning a larger fraction of the edges in $E'$ to the vertices in $U$. Since we maintain feasibility of $(\hat{x}, \hat{y})$ in LP 2.1, the properties of $\text{VC}$ still hold, but this allows us to derive a stronger bound when considering a random choice for the threshold.

4.1 An Improved Approximation Algorithm.
Our improved algorithm is parametrized by a probability distribution $\mathcal{A}$ for $\alpha$, where $\mathcal{A}$ satisfies $\mathbb{P}_\mathcal{A}(\alpha \in [0, 1/2]) = 1$. We now present the algorithm $\text{ALG}'_\mathcal{A}$ formally:

1. Sample a random threshold $\alpha$ according to the distribution $\mathcal{A}$.
2. Solve LP 2.1; let $(x^*, y^*)$ be an optimal solution.
3. Similar to Step 1 in algorithm VC, define a partition of $V$ into $U \cup W \cup Z$, and $E'$ as follows:
   - $U = \{u \in V : x_u^* \geq \alpha\}$,
   - $W = \{w \in V : 0 < x_w^* < \alpha\}$,
   - $Z = \{z \in V : x_z^* = 0\}$,
   - $E' = \{e = uw \in E : u \in U, w \in W\}$.
4. Modify $(x^*, y^*)$ to form $(\hat{x}, \hat{y})$ as follows:
   $$\hat{x}_u = \left\{ \begin{array}{ll}
   [x_u^*] & \text{if } v \in U \\
   0 & \text{otherwise},
   \end{array} \right.$$ 
   and
   $$\hat{y}(e, v) = \min \left\{ 1, y^*(e, u) \frac{[x_u^*]}{x_u^*} \right\},$$ 
   for $e = uw \in E'$, where $u \in U$ and $w \in W$, set
   $$\hat{y}(e, w) = 1 - \hat{y}(e, u),$$ 
   for $e \in E \setminus E'$ and $v \in e$, set $\hat{y}(e, v) = y^*(e, v)$.
5. Run the subroutine $\text{VC}(\hat{x}, \hat{y}, \alpha)$. Let $(x', y')$ be the returned solution.

First, to show that $\text{ALG}'_\mathcal{A}$ indeed returns a $\text{VCHC}$, it suffices to show that it is legitimate to input $(\hat{x}, \hat{y})$ to $\text{VC}$ in Step 5. It is justified by the following Lemma:

**Lemma 4.1.** The modified solution $(\hat{x}, \hat{y})$ in $\text{ALG}'_\mathcal{A}$ is feasible for LP 2.1.

Lemma 4.1 asserts that by rounding up $x_u^*$ to $\hat{x}_u$ for all $u \in U$, it gives enough room to scale up $y^*(e, u)$ to $\hat{y}(e, u)$ for $e \in \delta(u)$ for all $u \in U$, while satisfying the constraints in LP 2.1. The proof of Lemma 4.1 is elementary, and is deferred to Appendix A.

Next, we show that randomizing the threshold gives an approximation ratio better than 3 by choosing the following probability distribution $\mathcal{A}$:

$$\mathbb{P}_\mathcal{A}(\alpha \leq x) = \begin{cases} 
\frac{1 - \beta}{\beta} \frac{x}{1 - x} & \text{if } 0 \leq x \leq \beta \\
1 & \text{if } x > \beta.
\end{cases}$$

Here $\beta = 2\sqrt{3} - 3 \approx 0.464$. The rationale behind the choice of $\mathcal{A}$ is explained in the proof of the following theorem:

**Theorem 4.1.** Let $C$ be the cost of the output of $\text{ALG}'_\mathcal{A}$, where $\mathcal{A}$ is as defined in (4.5). Then $\mathbb{E}[C] \leq (1 + 2/\sqrt{3})\sum_{v \in V} x_v^*$. Hence, $\text{ALG}'_\mathcal{A}$ returns a $(1 + 2/\sqrt{3})$-approximation to $\text{VCHC}$ on multigraphs.

The proof of Theorem 4.1 can be found in Appendix B. The algorithm $\text{ALG}'_\mathcal{A}$ can be easily derandomized by trying for the threshold $\alpha$ all values in the set $\Gamma = \{x_v^* : x_v^* \leq \beta\}$, and by outputting the best solution. Indeed the thresholding for some $\alpha$ is unaffected if we increase $\alpha$ to the next value within $\Gamma$.

5 A 2f-approximation to VCHC on f-hypergraphs

In this section, we propose a $2/\alpha$-approximation algorithm $\text{ALG}^\alpha_f$ for $\text{VCHC}$ on $f$-hypergraphs, where $\alpha \in [0, 1/f]$, by extending the framework in Section 3. Recall that an $f$-hypergraph is one with the size of largest hyperedge at most $f$. Although setting $\alpha = 1/f$ gives the smallest approximation ratio, we keep the parametrization with $\alpha$ to maintain full generality. Now, the following is an LP relaxation of $\text{VCHC}$ on an $f$-hypergraph:

$$\min \sum_{v \in V} x_v$$

s.t.

(5.6a) $\sum_{w : w \in e} y(e, u) = 1 \quad \forall e \in E$

(5.6b) $y(e, u) \leq x_u \quad \forall e \in E, u \in e$

(5.6c) $\sum_{e \in \delta(u)} y(e, u) \leq k_u x_u \quad \forall u \in V$

(5.6d) $x_u \leq m_u \quad \forall u \in V$

(5.6e) $x, y \geq 0$
LP 5.6 has a similar interpretation as LP 2.1. Before presenting ALG_a^b, we note that when constructing a feasible solution \((x, y)\) to VCHC on hypergraphs, we only need the integrality of \(x\), but not \(y\).

**Lemma 5.1.** Suppose \((x, y)\) is a feasible solution for LP 5.6 with \(x\) integral. Then we can compute efficiently an integral \(y'\) such that \((x, y')\) is a feasible VCHC by a maximum flow computation.

Lemma 5.1 generalizes Lemma 2.1, and the proof is offered in Appendix C. Next, we state our approximation algorithm ALG_a^b to VCHC on \(f\)-hypergraphs below:

1. Solve LP 5.6 for an optimal solution \((x^*, y^*)\). Partition \(V\) into \(U \cup W \cup Z\), where:
   - \(U = \{u \in V : x_u^* \geq \alpha\}\),
   - \(W = \{w \in V : 0 < x_w^* < \alpha\}\),
   - \(Z = \{z \in V : x_z^* = 0\}\).

   Note that no hyperedge is completely contained in \(W \cup Z\). It is because \(e \in W \cup Z\) implies that
   
   \[1 = \sum_{v \in e} y(e, v) \leq \sum_{v \in e} x_v \leq f \max_{v \in e} x_v < 1,
   \]
   which is a contradiction.

2. Define the partial solution
   
   \[x'_w = \begin{cases} \left\lfloor x_u^* \right\rfloor & \text{if } u \in U \\ 0 & \text{if } u \in Z. \end{cases}\]

   Note that we have not defined \(x'_w\) for \(w \in W\) yet.

3. Similar to the case in multigraphs, the capacities provided by \(x_u^*\) for \(U\) are enough to cover the edges lying entirely within \(U \cup Z\). However, these capacities may not be enough to fully cover the edges in \(E' := \{e \in E : e \cap W \neq \emptyset\}\), where \(e \cap W\) denotes the set of vertices in \(W\) that are incident to \(e\). Thus we design a covering LP to select a subset of \(W\) into the vertex cover, in order to provide full coverage for \(E'\).

   For an edge \(e \in E'\), our partial solution \(x'\) covers a fraction \(\sum_{w \in W} y^*(e, w)\) of \(e\), but there might be a deficit of \(\sum_{w \in W} y^*(e, w)\) for the coverage of \(e\). Unlike the case of multigraphs, this deficit can be associated to more than one vertex in \(e \cap U\) when \(|e \cap U| > 1\). We distribute the deficit of \(e\) among \(e \cap U\) as follows. We associate with \(u\) a partial deficit of

   \[\gamma^e_u = \sum_{w \in W} y^*(e, w),\quad \text{where } \gamma^e_u = \frac{y^*(e, u)}{\sum_{v \in e \cap U} y^*(e, v)}.
   \]

   Note that \(0 \leq \gamma^e_u \leq 1\), and \(\sum_{v \in e \cap U} \gamma^e_v = 1\). Therefore, for each \(e \in E'\), its deficit is fully distributed among the vertices in \(e \cap U\). The total amount of deficit of \(u \in U\) is

   \[r(u) = \sum_{e \in \delta'(u)} \gamma^e_u \sum_{w \in W} y^*(e, w).
   \]

   Next, we consider the amount of coverage \(w \in W\) can provide. Suppose we include \(w\) into our cover. Then for \(e \in \delta(w)\) and \(u \in e \cap U\), \(w\) can offer a coverage of \(\gamma^e_u y^*(e, w)/x_w^*\) for \(u\), since we can scale up the coverage of \(e\) from \(w\) accordingly when we scale up \(x_w^*\) to 1. However, if we include a subset \(\{w_1, \ldots, w_s\} \subset e \cap W\) in our vertex cover, this subset can only offer a coverage of \(\min\{1, \sum_{i=1}^{s} \gamma^e_{w_i} y^*(e, w_i)/x_w^*\}\) for \(u\) on edge \(e\), since each edge needs at most 1 unit of coverage.

   To overcome this non-linearity in our formulation of the covering LP, we scale up the coverage \(\gamma^e_u y^*(e, w)/x_w^*\) to \(\gamma^e_u y^*(e, w)/(f - 1)x_w^*\). This resolves the issue of non-linearity, since the total amount of coverage of \(e \cap W\) on \(e\) for any \(u \in e \cap U\) is always less than or equal to 1. Now, vertex \(w\), if selected, can provide

   \[M(u, w) = \sum_{e \in \delta(w), u \in e} \gamma^e_u y^*(e, w)/x_w^* (f - 1),\]

   units of coverage to \(u \in U\). Altogether, the discussion motivates the following linear program:

   \[
   \begin{align*}
   \min & \sum_{w \in W} z_w \\
   \text{subject to} & \sum_{w \in W} M(u, w) z_w \geq r(u) & \forall u \in U \\
   & 0 \leq z_w \leq 1 & \forall w \in W.
   \end{align*}
   \]

   When \(f = 2\), it is in fact LP 3.2. Now, solve LP 5.7 for an extreme point solution \(z^*\).

4. Define \(x'_w = \left\lceil z^*_w \right\rceil\) for all \(w \in W\). Note that \(x'\) is now integral.

5. Compute an integral assignment \(y'\) via network flows as in Lemma 5.1, and output \((x', y')\).

**5.1 The Analysis of ALG_a^b.** First, we prove the following which shows the feasibility of the algorithm.

**Theorem 5.1.** Let \((\hat{x}, \hat{y})\) be feasible for LP 5.6, and \(0 \leq \alpha \leq 1/f\). Consider any feasible solution \(z\) to the
corresponding LP 5.7. Then for
\[ x_v = \begin{cases} [\hat{x}_v] & \text{if } v \in U \\ z_v & \text{if } v \in W \\ 0 & \text{if } v \in Z \end{cases} \]
there exists \( y \) such that \( (x, y) \) is feasible for LP 5.6.

The proof of Theorem 5.1 can be found in Appendix D. By putting \( (\hat{x}, \hat{y}) = (x^*, y^*) \) and \( z = [z^*] \), Theorem 5.1 shows that \( \text{ALG}_\alpha \) returns a feasible VCHC. Note that Theorem 5.1 generalizes Theorem 3.1.

Next, we establish the approximation guarantee of \( \text{ALG}_\alpha \). Now, the total cost of the solution \((x', y')\) is
\[ \sum_{v \in V} x'_v = \sum_{u \in U} [x^*_u] + \sum_{w \in W} [z^*_w]. \]

To bound the total cost, we first have the following bound on \( \sum_{w \in W} z^*_w \).

**Lemma 5.2.** The solution \( z_w = (f - 1)x^*_w \), for all \( w \in W \), is a feasible solution to LP 5.7. In particular, \( \sum_{w \in W} z^*_w \leq (f - 1)\sum_{w \in W} x^*_w \).

**Proof.** First, we have \( 0 \leq (f - 1)x^*_w \leq (f - 1)/f \leq 1 \) for every \( w \in W \). For every \( u \in U \), we have
\[ \sum_{w \in W} M(u, w)(f - 1)x^*_w = \sum_{w \in W \cap u, w \in e} \gamma^*_u y^*(e, w) x^*_w (f - 1) = \sum_{ed(u) \in e \cap W} \gamma^*_u y^*(e, w) = r(u). \]

Note that Lemma 5.2 generalizes Lemma 3.1. Finally, the following theorem establishes the approximation ratio of \( \text{ALG}_\alpha \).

**Theorem 5.2.** \( \text{ALG}_\alpha \) returns a \( 2/\alpha \)-approximation for the VCHC on hypergraphs. In particular, when \( \alpha = 1/f \), it returns a \( 2f \)-approximation.

**Proof.** Let \( W_f := \{ w \in W : z^*_w \in (0, 1) \} = \{ w_1, \ldots, w_r \} \) be the set of \( w \in W \) with fractional \( z^*_w \) values. In particular, we have \( |W_f| = r \). The total cost can be bounded as follows:
\[ \sum_{v \in V} x'_v = \sum_{u \in U} [x^*_u] + \sum_{w \in W} [z^*_w] \leq \frac{1}{\alpha} \sum_{u \in U} x^*_u + \sum_{w \in W} z^*_w + r \leq \frac{1}{\alpha} \sum_{u \in U} x^*_u + (f - 1) \sum_{w \in W} x^*_w + r. \]

The first inequality is by \( x^*_u \geq \alpha \) for every \( u \in U \), and the second is by Lemma 5.2. To bound \( r \), by the argument in Theorem 3.2, we know that there exists distinct vertices \( u_1, \ldots, u_r \in U \) such that \( \prod_{j=1}^r M(u_j, w_j) \neq 0 \). That means
\[ r \leq |U| \leq \frac{1}{\alpha} \sum_{u \in U} x^*_u. \]

Overall, the bound for the cost is
\[ \sum_{v \in V} x'_v \leq \frac{2}{\alpha} \sum_{u \in U} x^*_u + (f - 1) \sum_{w \in W} x^*_w \leq \frac{2}{\alpha} \sum_{v \in V} x^*_w. \]

When \( f = 2 \), Theorem 5.2 gives a guarantee of 4, while Theorem 3.3 gives a guarantee of 3. It is because in the argument for general hypergraphs, there is no matching structure associated with an extreme point solution of LP 5.7. Hence, this leads to a slightly coarser analysis.

**6 A Reduction from Partial VCHC to VCHC.**

In this section, we consider the Partial Vertex Cover with Hard Capacities constraint (Partial-VCHC), which is a generalization of VCHC. An instance of Partial-VCHC is specified by the tuple \( I = (V, E, k, m, \ell) \), and is defined as follows. We are provided with a VCHC instance with parameters \((V, E, k, m)\), but now we are only required to cover all but \( \ell \) edges, instead of covering all. In particular, when \( \ell = 0 \), it is a VCHC problem. We provide the following reduction from Partial-VCHC to VCHC:

**Theorem 6.1.** For a given integer \( f \geq 2 \), suppose we have an \( \eta_{f+1} \)-approximation algorithm to VCHC on \((f + 1)\)-hypergraphs. Then for any \( \epsilon > 0 \), there is an \( \eta_{f+1}(1 + \epsilon) \)-approximation algorithm to the Partial-VCHC on \( f \)-hypergraphs, which runs in \( \text{poly}(|V|^{1/\epsilon} |E|) \) time.

We prove Theorem 6.1 by reducing a Partial-VCHC instance on a \( f \)-hypergraph to a VCHC instance on an \((f + 1)\)-hypergraph.

**Proof.** Given a \( f \)-hypergraph Partial-VCHC instance \( I = (V, E, k, m, \ell) \), define a new \((f + 1)\)-hypergraph VCHC instance \( I' = (V', E', k', m') \), where
- \( V' = V \cup \{ s \} \), where \( s \) is a new vertex.
- \( E' = \{ e \cup \{ s \} : e \in E \} \). For each \( e \in E \), denote \( e' = e \cup \{ s \} \).
- \( k'_v = k_v \) if \( v \in V \), and \( k'_s = \ell \).
- \( m'_v = m_v \) if \( v \in V \), and \( m'_s = 1 \).
We compute a $\eta_{f+1}$-approximation $(x', y')$ to $I'$, and output the solution $(x, y)$, where $x$ is obtained by restricting $x'$ to $V$, and $y(e, v) = y'(e', v)$ for all $e \in E, v \in e$. Note that $(x, y)$ is feasible to $I$, since the total number of edges assigned is
\[
\sum_{e \in E} \sum_{v \in V, v \in e} y(e, v) = \sum_{e' \in E'} \sum_{v \in V, v \in e'} y'(e', v) = |E'| - \sum_{e' \in E'} y'(e, s) \geq |E| - \ell,
\]
so that $(x, y)$ covers all but at most $\ell$ edges. Next, we bound the cost as follows:
\[
\sum_{v \in V} x_v \leq \sum_{v \in V'} x_v' \leq \eta_{f+1}\text{opt}(I') \leq \eta_{f+1}\text{opt}(I) + 1 = \eta_{f+1} \left(1 + \frac{1}{\text{opt}(I)}\right) \text{opt}(I).
\]
Here, $\text{opt}(I)$ denotes the optimal value for instance $I$. To justify the third inequality above, we argue that $\text{opt}(I) + 1 \geq \text{opt}(I')$. It is because given an optimal solution $(x^*, y^*)$ to $I$, including $s$ in the cover and assigning the uncovered edges to $s$ will give a feasible solution to $I'$.

Lastly, to obtain the stated approximation ratio, we perform the following. For each $1 \leq C \leq 1/\epsilon$, and for each multisets of vertices of total size $C$, we check if it is feasible to $I$. This can be done by checking if $x_v \leq m_v$ for all $v$ and checking if the network flow instance defined in Lemma 5.1 has optimal value $\geq |E| - \ell$. If it is the case, output the multi-set along with the assignment defined by the optimal flow. This is the optimal Partial-VCHC. Otherwise, we know that $\text{opt}(I) > 1/\epsilon$, and the approximation algorithm above will have an approximation ratio $\leq \eta_{f+1}(1 + \epsilon)$. \hfill \square

Combining with ALG$_{1/f}^h$, we have the following:

**Corollary 6.1.** For $f \geq 2$, there is a $(2f + 2)(1 + \epsilon)$-approximation algorithm to the partial-VCHC problem on $f$-hypergraphs which runs in poly($|V|^{1/\epsilon}|E|$) time.

## 7 Conclusion

In summary, we have proposed new approximation algorithms to the minimum Vertex Cover with Hard Capacities problem (VCHC), which improve the approximation ratio from 38 to $1 + 2/\sqrt{3} < 2.155$ for multigraphs, and from max{65, 6f} to 2f for $f$-hypergraphs. Furthermore, we have presented a $(2f + 2)(1 + \epsilon)$-approximation for partial VCHC on $f$-hypergraphs. To conclude, we leave the open problem of whether there is a 2-approximation to the VCHC on multigraphs.

## References


we know from (A.1) that
\[ \hat{y}(e, w) \leq y^*(e, w) \leq x^*_w = \hat{x}_w. \]

For \( u \in U \), we know that
\[ \hat{y}(e, u) \leq 1 \leq [x^*_u] = \hat{x}_u. \]

For \( z \in Z \), we have \( \hat{y}(e, z) = 0 = \hat{x}_z \).

Finally, for prove the constraint (2.1c), for \( w \in W \), we have from (A.1) that
\[ \sum_{e \in \delta(w)} \hat{y}(e, w) \leq \sum_{e \in \delta(w)} y^*(e, w) \leq k_u x^*_w = k_u \hat{x}_w. \]

For \( u \in U \), we have
\[ \sum_{e \in \delta(u)} \hat{y}(e, u) \leq \frac{[x^*_u]}{x^*_u} \sum_{e \in \delta(u)} y^*(e, u) \leq \frac{[x^*_u]}{x^*_u} k_u x^*_u = k_u \hat{x}_u. \]

Finally, for \( z \in Z \), we have \( \sum_{e \in \delta(z)} \hat{y}(e, z) = 0 = k_z \hat{x}_z \).

Altogether, \((\hat{x}, \hat{y})\) is feasible to LP 2.1. \( \square \)

**B Proof of Theorem 4.1**

For a realization of the random threshold \( \alpha \in [0, \beta] \), (3.3) and Lemma 3.1 imply that the cost \( C \) of \( \text{ALG}'_A \) has the following upper bound:
\[
C \leq \sum_{v \in V} \{ [x^*_v] 1(x^*_v \geq \alpha) + x^*_v 1(x^*_v < \alpha) \\
+ 1(v \in U, v \text{ is in the matching } M) \}.
\]

Here, the matching \( M \) refers to the matching between \( W' \) and \( U \) identified in Theorem 3.2. Taking expectation on \( \alpha \) over the probability distribution \( A \) defined in (4.5), we have
\[
\mathbb{E}[C] \leq \sum_{v \in V} \{ [x^*_v] \mathbb{P}(v \text{ in the matching } M) \}.
\]

First, we provide the following bound for the third term:

**Lemma B.1.** The probability \( \mathbb{P}(v \in U, v \text{ is in } M) \) has the following upper bound:
\[
\leq f(x^*_v) := \begin{cases} 
\mathbb{P}(\alpha > 1 - x^*_v) & \text{if } 0 \leq x^*_v \leq 1 \\
\mathbb{P}(\alpha > 1 - \frac{x^*_v}{2}) & \text{if } x^*_v > 1
\end{cases}
\]

We note that the stronger bound derived in the range of \( x^*_v > 1 \) crucially uses the modification step (i.e. Step 4) in \( \text{ALG}'_A \). Also, note that the bound only depends on the value of \( x^*_v \). The proof of Lemma B.1 is given at the end of this appendix section.

Given the bound in Lemma B.1, we prove the following bound for the expected cost charged to each vertex:

**Lemma B.2.** For all \( v \in V \), we have the following bound:
\[
(B.2) \quad [x^*_v] \mathbb{P}(x^*_v \geq \alpha) + x^*_v \mathbb{P}(x^*_v < \alpha) + f(x^*_v) \leq \frac{x^*_v}{\beta}.
\]

Lemma B.2 immediately implies that \( \text{ALG}'_A \) is a \( 1/\beta < 2.155 \)-approximation algorithm. Our choice of the probability distribution \( A \) was engineered so as to minimize the factor in the above lemma.

**Proof.** [Proof of Lemma B.2] When \( x^*_v = 0 \) or \( x^*_v \geq 2 \), the bound (B.2) is clearly true. The main cases are when \( 0 < x^*_v \leq 1 \) and \( 1 < x^*_v < 2 \), as analyzed below.

When \( 0 < x^*_v \leq 1 \), the bound (B.2) is equal to the following:
\[
[x^*_v] \mathbb{P}(x^*_v \geq \alpha) + x^*_v \mathbb{P}(x^*_v < \alpha) + f(x^*_v) = \begin{cases} 
1 + 0 + \left( 1 - \frac{1 - \beta 1 - x^*_v}{x^*_v} \right) & \text{if } \beta < x^*_v \leq 1
\end{cases}.
\]

In each of the two cases, the bound is less than or equal to \( x^*_v / \beta \), and in fact for the case \( 0 \leq x^*_v \leq \beta \) it is an equality. Also, it is easy to check that our choice of \( A \) minimizes the factor in the above analysis.

When \( 1 < x^*_v < 2 \), the bound (B.2) is equal to the following:
\[
[x^*_v] \mathbb{P}(x^*_v \geq \alpha) + x^*_v \mathbb{P}(x^*_v < \alpha) + f(x^*_v) = 2 + 0 + \mathbb{P}(\alpha \geq 1 - x^*_v / 2) = 2 + \left( 1 - \frac{1 - \beta 2 - x^*_v}{x^*_v} \right) < \frac{x^*_v}{\beta},
\]

where the last inequality is true for all \( x^*_v > 1 \). \( \square \)

Finally we return to the proof of Lemma B.1:

**Proof.** [Proof of Lemma B.1] We first show that the bound
\[
\mathbb{P}(v \in U, v \text{ is in } M) \leq \mathbb{P}(\alpha > 1 - x^*_v)
\]
holds for all \( v \in V \). Indeed, if \( v \in U \) is covered by an edge \( vw \in M \), then we know that \( w \in W \), which implies that
\[
1 = y^*(e, v) + y^*(e, w) \leq x^*_v + x^*_w < x^*_v + \alpha.
\]

Next, we show the stronger bound
\[
\mathbb{P}(v \in U, v \text{ is in } M) \leq \mathbb{P}(\alpha > 1 - \frac{x^*_v}{2})
\]
when \( x^*_v > 1 \). Since it is clearly true when \( x^*_v \geq 2 \), we focus on the range \( 1 < x^*_v < 2 \). Now, if \( v \in U \) is in the matching, then there must exist \( w \) such that
\[
M(v, w) = \sum_{e : e = vw} \max \left\{ 0, 1 - y^*(e, v) \frac{x^*_v}{x^*_v} \right\} \cdot \frac{1}{x^*_w} > 0.
\]
Take an edge \( e \) between \( v \) and \( w \) such that \( \max \{0, 1 - y^*(e, w) \frac{x_w^*}{x_v^*}\} \cdot \frac{1}{x_v^*} > 0 \). Since \( 1 < x_v^* < 2 \), it implies that
\[
1 > y^*(e, v) \frac{x_v^*}{x_v^*} = y^*(e, v) \frac{2}{x_v^*}.
\]
Finally, we have
\[
x_v^* > 2y^*(e, v) = 2(1 - y^*(e, w)) \geq 2(1 - x_w^*) \geq 2(1 - \alpha)
\]
as desired. \( \square \)

**C Proof of Lemma 5.1**

Given an instance \( I = (V, E, k, m) \), and a solution \( (x, y) \) to LP 5.6, where \( x \) is integral, construct the following network flow instance:

- The set of nodes is \( s \cup V \cup E \cup t \), where \( s \) is the source, \( t \) is the sink.
- The set of arcs and their capacities are defined as follows:
  - For all \( v \in V \), there is an arc from \( s \) to \( v \) with capacity \( k_v \).
  - For all \( v \in V \) and \( e \supseteq v \), there is an arc from \( v \) to \( e \) with capacity \( x_v \).
  - For all \( e \in E \), there is an arc from \( e \) to \( t \) with capacity 1.

Now, the feasibility of \( (x, y) \) for LP 5.6 implies that there is a flow of value \(|E|\) from \( s \) to \( t \) in the instance \( I \). But since all capacities in \( I \) are integral, there also exist an integral flow of value \(|E|\) from \( s \) to \( t \). Now, for all \( e \in E \), \( v \in e \), define \( y'(e, v) = 1 \) if there is one unit flow through the arc \((v, e)\), and \( y'(e, v) = 0 \) otherwise. Then \((x, y')\) is a feasible solution to VCHC on instance \( I \). \( \square \)

**D Proof of Theorem 5.1**

In the proof, we define a \( y \) such that \((x, y)\) is feasible as follows:

- For \( e \in E' \) and \( v \in e \), we define:
  - If \( v \in e \cap W \), \( y(e, v) = \frac{\hat{y}(e, v)}{(f - 1)\hat{x}_w}z_w \).
  - If \( v \in e \cup U \),
    \[
y(e, v) = \gamma_v^u \left( 1 - \sum_{w \in e \cap W} \frac{\hat{y}(e, w)}{(f - 1)\hat{x}_w}z_w \right).
    \]
  - If \( v \in e \cap Z \), \( y(e, v) = 0 \).
- For \( e \in E - E' \), for all \( u \in e \), define \( y(e, u) = \hat{y}(e, u) \).

We claim that \((x, y)\) is feasible to LP 5.6.

First, (5.6a) is satisfied. For \( e \in E - E' \), it clearly holds. For \( e \in E' \), we have
\[
\sum_{v \in e} y(e, v) = \sum_{u \in e \cap U} \gamma_v^u \left( 1 - \sum_{w \in e \cap W} \frac{\hat{y}(e, w)}{(f - 1)\hat{x}_w}z_w \right) + \sum_{w \in e \cap W} \frac{\hat{y}(e, w)}{(f - 1)\hat{x}_w}z_w = 1,
\]
since \( \sum_{e \cap W} \gamma_v^u = 1 \).

Secondly, (5.6d) is clearly satisfied, since for \( u \in U \), \( \hat{x}_u \leq m_u \) implies that \( x_u = \lceil \hat{x}_u \rceil \leq m_u \), and for \( w \in W \), \( x_w \leq 1 \leq m_w \), where the latter inequality is due to \( \hat{x}_w > 0 \). For \( z \in Z \), we have \( x_z = 0 \leq m_z \).

Thirdly, (5.6e) is satisfied. For the \( x \) variables, we know that they are non-negative. For the \( y \) variables, we have the following cases. For \( e \in E - E' \), it is clear. For \( y(e, w) \) where \( e \in E', w \in W \), it is also clear. For \( y(e, u) \) where \( e \in E', u \in U \), we have
\[
y(e, u) = \gamma_v^u \left( 1 - \sum_{w \in e \cap W} \frac{\hat{y}(e, w)}{(f - 1)\hat{x}_w}z_w \right) \geq \gamma_v^u \left( 1 - \sum_{w \in e \cap W} \frac{1}{(f - 1)} \right) \geq 0,
\]
since \( \hat{y}(e, w)/\hat{x}_w \leq 1 \). We note that in showing (D.3), we crucially use the fact that the contribution \( \frac{\hat{y}(e, w)}{\hat{x}_w} \) is scaled down to \( \frac{\hat{y}(e, w)}{(f - 1)\hat{x}_w} \).

Fourthly, (5.6b) is satisfied. For \( w \in W \), we have
\[
y(e, w) = \frac{\hat{y}(e, w)}{(f - 1)\hat{x}_w}z_w \leq \frac{z_w}{f - 1} \leq z_w.
\]
For \( u \in U \), we first define the notation \( \delta'(u) = \{ e \in \delta(u) : e \cap W \neq \emptyset \} \). Going back to checking (5.6b) for \( u \), if \( e \in \delta'(u) \setminus \delta(u) \), it is clearly true. If \( e \in E', y(e, u) \leq x_u \), since
\[
y(e, u) \leq 1 - \sum_{w \in e \cap W} \frac{\hat{y}(e, w)}{(f - 1)\hat{x}_w}z_w \leq 1 \leq x_u.
\]
For \( z \in Z \), we have \( y(e, v) = x_v = 0 \).

Finally, it remains to check (5.6c). For \( w \in W \), we have
\[
\sum_{e \in \delta(w)} y(e, w) = \sum_{e \in \delta(w)} \frac{\hat{y}(e, w)}{(f - 1)\hat{x}_w}z_w \leq \frac{k_w x_w}{(f - 1)\hat{x}_w}z_w \leq k_w z_w = k_w x_w.
\]
For $u \in U$, to check that its corresponding constraint in (5.6c) is satisfied, we first claim that
\[
\sum_{e \in \delta'(u)} y(e, u) \leq \sum_{e \in \delta'(u)} \hat{y}(e, u).
\]

Given the claim, we can conclude that
\[
\sum_{e \in \delta(u)} y(e, u) = \sum_{e \in \delta'(u)} y(e, u) + \sum_{e \in \delta(u) \setminus \delta'(u)} y(e, u)
\]
\[
= \sum_{e \in \delta'(u)} y(e, u) + \sum_{e \in \delta(u) \setminus \delta'(u)} \hat{y}(e, u)
\]
\[
\leq \sum_{e \in \delta'(u)} \hat{y}(e, u) + \sum_{e \in \delta(u) \setminus \delta'(u)} \hat{y}(e, u)
\]
\[
\leq k_u \hat{x}_u \leq k_u x_u.
\]

Finally, to verify the claim, we have:
\[
\sum_{e \in \delta'(u)} y(e, u) = \sum_{e \in \delta'(u)} \gamma^e_u \left( 1 - \sum_{w \in \epsilon \setminus W} \hat{y}(e, w) \frac{(f - 1)\hat{x}_w}{z_w} \right)
\]
\[
= \sum_{e \in \delta'(u)} \gamma^e_u - \sum_{w \in W} \sum_{e \in E' \setminus z \in W} \gamma^e_u \hat{y}(e, w) \frac{(f - 1)\hat{x}_w}{z_w}
\]
\[
= \sum_{e \in \delta'(u)} \gamma^e_u - \sum_{w \in W} M(u, w) z_w
\]
\[
\leq \sum_{e \in \delta'(u)} \gamma^e_u - r(u)
\]
\[
= \sum_{e \in \delta'(u)} \gamma^e_u - \sum_{e \in \delta'(u)} \gamma^e_u \sum_{w \in \epsilon \setminus W} \hat{y}(e, w)
\]
\[
= \sum_{e \in \delta'(u)} \gamma^e_u \sum_{v \in \epsilon \setminus U} \hat{y}(e, v) = \sum_{e \in \delta'(u)} \hat{y}(e, u).
\]

(D.4)

We crucially use the definition of $\gamma^e_u$ in (D.4) to establish (5.6c) for $u \in U$. Finally, for $z \in Z$, we have
\[
\sum_{e \in \delta(z)} y(e, z) = 0 = k_z x_z. \quad \text{Thus, constraint (5.6c) is satisfied for all } v \in V. \quad \text{Altogether, } (x, y) \text{ is feasible to LP 5.6.} \]