Approximate max-integral-flow/min-multicut theorems

Kenji Obata*
Computer Science Division
University of California, Berkeley
Berkeley, CA 94720
kenjioba@eecs.berkeley.edu

ABSTRACT

We establish several approximate max-*integral*-flow / min-multicut theorems. While in general this ratio can be very large, we prove strong approximation ratios in the case where the min-multicut is a constant fraction ϵ of the total capacity of the graph. This setting is motivated by several combinatorial and algorithmic applications. Prior to this work, a general max-integral-flow / min-multicut bound was known only for the special case where the graph is a tree. We prove that, for arbitrary graphs, the max-integral-flow / min-multicut ratio is $O(\epsilon^{-1} \log k)$, where k is the number of commodites; for graphs excluding a fixed subgraph as a minor (for instance, planar graphs), $O(1/\epsilon)$; and, for dense graphs, $O(1/\sqrt{\epsilon})$. Our proofs are constructive in the sense that we give efficient algorithms which compute either an integral flow achieving the claimed approximation ratios, or a witness that the precondition is violated.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*Computations on discrete structures*; G.1.6 [**Numerical Analysis**]: Optimization—*Linear programming*; G.2.2 [**Discrete Mathematics**]: Graph Theory—*Path and circuit problems*

General Terms

Algorithms, Theory

1. INTRODUCTION

A classic theorem of Ford and Fulkerson asserts that the maximum flow between two vertices in a graph equals the weight of a minimum cut separating the two vertices [6]. This flow/cut duality further has a beautiful *integrality* property: If the capacities of the edges of G are integers, then there exists an integral flow achieving the weight of the minimum cut. Integrality is often a necessary or desirable property in applications. In this work, we consider corresponding integrality results in the setting of *multicommodity* flows.

Unfortunately, the analogous equality does not generally hold in the multicommodity setting. However, in a body of work initiated by Leighton and Rao ([17, 19, 7, 13, 1, 18] and others), a number of *approximate* max-flow / min-cut relations have been established. All of these results differ in a fundamental way from the Ford-Fulkerson theorem in that they are based on dual rounding procedures. This approach loses the guarantee of primal integrality and, indeed, the integrality gap for multicommodity flows can be $\Omega(k)$, where k is the number of commodities [8].

In this work, we establish several approximate maximum-*integral*-flow/minimum-multicut theorems. While in general this ratio can be very large, we prove strong approximation ratios in the case where the min-multicut is a constant fraction ϵ of the total capacity of the graph. This setting is motivated by several combinatorial and algorithmic applications. Prior to this work, a general max-integral-flow / min-multicut bound was known only for the special case where the graph is a tree [8]. We prove that, for arbitrary graphs, the max-integral-flow / min-multicut ratio is $O(\epsilon^{-1} \log k)$; for graphs excluding a fixed subgraph as a minor (for instance, planar graphs), $O(1/\epsilon)$; and, for dense graphs, $O(1/\sqrt{\epsilon})$. Our proofs are constructive in the sense that we give efficient algorithms which compute either an integral flow achieving the claimed approximation ratios, or a witness that the precondition is violated.

Our main results are summarized in the following theorem.

THEOREM 1. *Let $G(V, E)$ be a graph with capacity function $c : E \to \mathbf{Z}^+$, and K be a demand graph on a k-element subset of V such that the weight of any multicut separating all pairs of vertices $(k_1, k_2) \in K$ is at least ϵC, where $C = \sum_{e \in E} c(e)$. Then*

(i) For any G, the max-integral-flow/min-multicut ratio is $O(\epsilon^{-1} \log k)$. If k^* is the vertex cover number of K, the ratio is improved to $O(\epsilon^{-1} \log k^*)$.

(ii) If, for some constant r, G does not contain $K_{r,r}$ as a minor (for instance, if G is planar), then the ratio is improved to $O(1/\epsilon)$.

(iii) If $c \in \{0, 1\}^E$ and G is δ-dense for some $\delta > 0$, then the ratio is improved to $O(1/\sqrt{\epsilon\delta})$.

*Work supported by an NSF graduate fellowship.

STOC'04, June 13–15, 2004, Chicago, Illinois, USA.

The proof of Theorem 1, as in previous work in this area, depends on efficient low-radius decompositions in a graph metric. The traditional approach is to decompose G using the metric induced by a minimum solution to the dual of the maximum flow problem; the approximation ratio follows by the strong duality theorem. The fundamental difference in this work is that G is decomposed according to the *unweighted* distance metric on G, and this is used to construct a large *primal* solution; the ratio follows by the promised lower bound on the minimum multicut. Two of the decompositions we use are standard in the multicommodity flow literature, while the third is a "new" one. Although we reuse existing decomposition lemmas, our analysis works for fundamentally different reasons than in the dual rounding procedures for which these decompositions were originally designed. The details are discussed in Section 2.

The approximate maximum integral flow algorithm works roughly as follows: For a given family of graphs (general, planar, dense, etc.), we define a "greed" function $g(t)$ which is an increasing function of time t. At time t, the algorithm greedily increases the primal value by pushing a unit of flow along a path of length at most $g(t)$. If this can repeated for sufficiently many iterations then we are done. Otherwise, the algorithm constructs a witness multicut by combining both primal and dual information. If one assumes that the min-multicut precondition is satisfied, then this algorithm can be viewed as a pure greedy algorithm which always pushes a unit of flow along the shortest available path between any terminal pair. The details are presented in Section 3.

Greedy algorithms have been investigated in the context of the closely related edge-disjoint paths (EDP) problem. Kleinberg [14] studied an on-line variant of the greedy algorithm with $g(t) = L$ fixed, and showed that this constructs a set of edge-disjoint paths of size within a factor $O(\max(\mathrm{diam}(G), m^{1/2}))$ of the optimum, where m is the number of edges. Kolliopoulos and Stein [15] proved that the pure greedy algorithm achieves an approximation ratio of $O(d_0)$, where d_0 is the average length of paths in any optimal solution. Chekuri and Khanna [4] subsequently proved an approximation ratio of $O(n^{2/3})$ with respect to the number n of vertices. More recently, Varadarajan and Venkataraman [20] proved a ratio of $O(n^{2/3} \log^{2/3} n)$ applicable to the more general case of directed graphs. Theorem 1 shows that the greedy algorithm achieves much stronger (logarithmic- or constant-factor) approximation ratios to the maximum EDP and related maximum integral flow problems, subject to appropriate conditions on the problem instance. These and other algorithmic and combinatorial applications of Theorem 1 are described in Section 4.

1.1 Definitions and notation

Throughout, $G(V, E)$ denotes a simple, undirected graph, $c : E \to \mathbf{Z}^+$ denotes a $\mathbf{Z}^+$-valued capacity function on the edges of G, and K denotes a simple, undirected, unweighted graph defined on a vertex-subset of V. A *K-path* in G is a simple path in G between k_1, k_2 where $(k_1, k_2) \in E(K)$, a *K-cut* is an edge-subset $F \subseteq E$ such that $G \setminus F$ has no K-path, and a *K-partition* is a disjoint system of vertex-subsets $(S_1, \ldots, S_t)$ such that each $k \in K$ is in some S_i.

By the *characteristic vector* of a simple path P, we mean the vector in $\mathbf{Z}^E$ which is 1 on edges $e \in P$ and 0 otherwise. A set $\{p_1, \ldots, p_t\}$ of characteristic vectors of K-paths along with positive weights $w_1, \ldots, w_t$ is called a *K-flow* with respect to a capacity function c when $v = \sum_i w_i p_i$ is such that $v(e) \leq c(e)$ for all edges e. The *weight* of a K-flow is the sum $\sum w_i$ of all path weights. By a maximum flow, we mean a K-flow with maximum weight among all K-flows. A flow is *integral* when all path weights w_i are integer-valued. A maximum *integral* flow is a flow with maximum weight among all integral K-flows.

Often, we identify c with the graph formed by including edges for which c is non-zero. When $G(V, E)$ is understood, we write (S_1, S_2) to denote the subset of edges with one endpoint in S_1 and the other in S_2, where $S_i \subseteq V$. For $F \subseteq E$, we define $c(F) = \sum_{e \in F} c(e)$, and for $W \subseteq V, c(W) = c((W, W))$. Throughout,

$$C = c(E).$$

If $S \subseteq V$, we write $\nabla(S) = (S, V \setminus S)$ and

$$\nabla(S_1, \ldots, S_t) = \bigcup_i \nabla(S_i).$$

For a capacity function c, $\nabla_c(S)$ denotes the capacity function which equals c on $\nabla(S) \cap \{e : c(e) > 0\}$ and is zero elsewhere. By the norm $\|c\|$ of c we mean $\|c\|_1 = \sum_e c(e)$.

We use $d(v_1, v_2)$ to denote the length of a shortest path between vertices v_1 and v_2 in G,

$$B(v, \rho) = \{w \in V \mid d(v, w) \leq \rho\}$$

to denote the closed ball of radius ρ around v and

$$B^\circ(v, \rho) = \{w \in V \mid d(v, w) = \rho\}$$

to denote its boundary. The diameter

$$\mathrm{diam}(S) = \max_{v_i, v_j \in S} d(v_i, v_j)$$

and the radius $\rho(S) = \frac{1}{2}\mathrm{diam}(S)$. The radius of a K-partition $(S_1, \ldots, S_t)$ is just the maximum over i of $\rho(S_i)$.

2. THREE GRAPH DECOMPOSITIONS

We discuss three low-radius graph decomposition algorithms. The first is generic but gives the weakest bounds; the latter two apply only to specialized cases, but give much stronger results. The first two have been used previously to establish max-flow / min-multicut approximation ratios, while the third one is "new".

For all $\rho \geq 0$ and families $\mathcal{G}$ of $\mathbf{Z}^+$-valued edge-functions on graphs, define

$$f_{\mathcal{G},k}(\rho) = \max_{\substack{c \in \mathcal{G}, C > 0 \\ K \subseteq V, |K| \leq k}} \frac{1}{C} \left(\min_{(S_1, \ldots, S_t) \in \mathcal{P}_K^\rho} c\left(\nabla(S_1, \ldots, S_t)\right) \right)$$

where $\mathcal{P}_K^\rho$ denotes the set of all K-partitionings of V of radius ρ. For brevity, we will omit the parameter k. Note that, for any $\mathcal{G}$, $f_{\mathcal{G}}(\rho)$ is monotonically decreasing in ρ since, for $\rho' > \rho$, a radius ρ decomposition is also a radius ρ' decomposition. Each of the following decompositions gives an algorithmic upper bound on $f_{\mathcal{G}}(\rho)$ for particular values of $\mathcal{G}$.

2.1 General graphs

We begin with the most generic decomposition algorithm, which is applicable to arbitrary graphs. The essential idea behind this decomposition was introduced by Leighton and

Rao in their original work [17]. The version which follows is the special case of a subsequent refinement by Garg, Vazirani, and Yannakakis [7] in which the dual variables are fixed to $(1, \ldots, 1)$.

"Garg-Vazirani-Yannakakis" Decomposition: Let $\alpha > 0$ be a parameter, to be selected later. While there exists any terminal vertex v in G repeat the following: Set $v \leftarrow$ an arbitrary vertex in $K \cap G$, $t \leftarrow 0$; while $c(\nabla(B(v,t))) + \frac{C}{k} > \alpha c(B(v,t))$, set $t \leftarrow t+1$; output $B(v,t)$ and set $G \leftarrow G \setminus B(v,t)$. The set of output subsets gives a K-partition of V.

LEMMA 2. *Let $S_1, \ldots, S_t$ denote the K-partition produced by the Garg-Vazirani-Yannakakis algorithm with parameter α. Then*

$$\rho(S_i) \leq \frac{\ln(k+1)}{\alpha} \quad \text{and} \quad c(\nabla(S_1, \ldots, S_t)) \leq 2\alpha C.$$

PROOF. Apply Lemmas 4.1 and 4.2 from [7] in the special case where $d = (1, \ldots, 1)$. □

COROLLARY 3. *Let $\mathcal{G}^*$ denote the set of all $\mathbf{Z}^+$-valued capacity functions. Then*

$$f_{\mathcal{G}^*}(\rho) \leq \frac{2\ln(k+1)}{\rho}.$$

2.2 Graphs excluding $K_{r,r}$

We now consider the special case of graphs excluding a fixed minor. This includes, for instance, the family of planar graphs. An ingenuous decomposition algorithm for this setting was designed by Klein, Plotkin, and Rao [13]. The version below is a variant in which, as before, we fix the input dual variables to $(1, \ldots, 1)$. We also scan along possible cut points to assure the cut weight on each iteration is at most a fixed fraction of the total capacity.

"Klein-Plotkin-Rao" Decomposition: Let $\alpha > 0$ be a parameter, to be selected later, and let r be such that $K_{r,r}$ does not occur as a minor of G. While G is non-empty, repeat the following: Set $v \leftarrow$ an arbitrary vertex in G and $t \leftarrow$ a value from $\{0, \ldots, \alpha - 1\}$ to be selected later; for $i = 1, \ldots, n$ let $G_i = \bigcup_{q \in R_i} B^\circ(v,q)$ where R_i is the ith set in the sequence $[0,t), [t, t+\alpha), [t+\alpha, t+2\alpha), \ldots$, and recurse on G_i to a maximum depth r. For each level of the recursion, select a value t such that the weight of the edges cut at that level is $\leq C/\alpha$ (such a value must exist because the cuts induced by each possible selection of t partition E into α classes). Interpret the set of regions at the bottom of the recursion as a vertex-partition of G, disregarding empty regions.

LEMMA 4. *Suppose that G does not contain $K_{r,r}$ as a minor and let $S_1, \ldots, S_t$ denote the vertex-partition produced by the Klein-Plotkin-Rao algorithm. Then*

$$\rho(S_i) \leq 2r^2\alpha \text{ and } c\left(\nabla(S_1, \ldots, S_t)\right) \leq \frac{r}{\alpha}C.$$

PROOF. In [13], it is proved that $\rho(S_i) \leq 2r^2\alpha$. By the "scanning" selection of t, we introduce a cut of weight at most C/α at each level of the recursion. Since the recursion has depth at most r, the resulting cut must have weight at most $\frac{r}{\alpha}C$. □

COROLLARY 5. *Let $\mathcal{G}^{r,r}$ denote the set of $\mathbf{Z}^+$-valued capacity functions on graphs excluding $K_{r,r}$. Then*

$$f_{\mathcal{G}^{r,r}}(\rho) \leq \frac{2r^3}{\rho}.$$

2.3 Dense graphs

For a fixed constant $0 < \delta \leq 1$, an unweighted graph is called *δ-dense* if $|E| \geq \delta n^2$. In other words, a graph is "dense" when a constant fraction of all possible edges are included in the graph. We achieve our tightest bounds when the capacity function c is 0-1-valued and G is δ-dense.

We use a technique due to Komlós which was originally introduced to prove tight bounds on the size of a minimum edge-set intersecting all odd cycles in a graph (see Section 4.2 below). We observe that this method is applicable in the much more general context of multicommodity flows. We also provide a new and simpler proof of Komlós' result.

"Komlós" Decomposition: Let $\alpha > 0$ be a parameter, to be selected later. While G is non-empty, repeat the following: Set $v \leftarrow$ an arbitrary vertex in G, $t \leftarrow 0$; while $|B^\circ(v,t)||B^\circ(v,t+1)| > \alpha|B(v,\infty)||B(v,t)|$ set $t \leftarrow t+1$; output $B(v,t)$ and set $G \leftarrow G \setminus B(v,t)$. The set of output subsets gives a partitioning of V.

LEMMA 6. *Let G be a δ-dense graph, and let $S_1, \ldots, S_t$ denote the vertex-partition produced by the Komlós algorithm. If $c \in \{0,1\}^E$, then*

$$\rho(S_i) \leq \frac{12}{\sqrt{\alpha}} \text{ and } c\left(\nabla(S_1, \ldots, S_t)\right) \leq \frac{\alpha C}{\delta}.$$

(In [16], Komlós achieves the better constant $\sqrt{2e}$ in place of 12, although our constant is not optimized.)

PROOF. For brevity, set $b_i = |B^\circ(v,i)|$ and $B_i = |B(v,i)|$, so that $B_i = \sum_{0 \leq j \leq i} b_j$. The stopping rule requires that the sequence $\{b_i\}$ satisfy $b_i b_{i+1} > \alpha n B_i$. Then, for each i, at least one of b_i, b_{i+1} must be at least $\sqrt{\alpha n B_i}$. So, consider alternating entries of $\{B_i\}$, $B'_j = B_{2j+1}$. Then the sequence $\{B'_j\}$ satisfies the recurrence

$$B'_0 \geq \alpha n \; ; \; B'_{j+1} \geq B'_j + \sqrt{\alpha n B'_j}.$$

We claim that $B'_j \geq \frac{\alpha n}{9} j^2$. This is verified by observation for $j = 0, 1$. For larger j, induction shows that

$$B'_{j+1} \geq \frac{\alpha n}{9} j^2 + \sqrt{\alpha n \frac{\alpha n}{9} j^2} = \frac{\alpha n}{9} j^2 + \frac{\alpha n}{3} j \geq \frac{\alpha n}{9}(j+1)^2$$

when $j \geq 1$. Then for odd i,

$$B_i \geq \frac{\alpha n}{9}\left(\frac{i-1}{2}\right)^2$$

so for $i \geq 2$,

$$B_i \geq \frac{\alpha n}{9}\left(\frac{i-2}{2}\right)^2 = \frac{\alpha n}{36}(i-2)^2$$

and (crudely), since $i - 2 \geq \frac{i}{2}$ for $i \geq 4$ and by inspection for $i = 0, \ldots, 3$, for any i,

$$B_i \geq \frac{\alpha n}{36}\left(\frac{i}{2}\right)^2 = \frac{\alpha n}{144} i^2.$$

But $B_i \leq n$ which implies that the maximum possible index i in such a sequence, which is also the maximum possible radius in a region, is at most $\sqrt{\frac{144}{\alpha}} = \frac{12}{\sqrt{\alpha}}$.

For the second part of the lemma, notice that the stopping rule implies that $c(\nabla(S_i)) \leq \alpha |S_i| n$. Then

$$\begin{aligned} c(\nabla(S_1, \ldots, S_t)) &\leq \sum_i c(\nabla(S_i)) \\ &\leq \sum_i \alpha |S_i| n \leq \alpha n^2 \leq \alpha C/\delta. \end{aligned}$$

□

COROLLARY 7. *Let $\mathcal{G}^\delta$ denote the set of δ-dense graphs with capacity functions $c \in \{0,1\}^E$. Then*

$$f_{\mathcal{G}^\delta}(\rho) \leq \frac{144}{\delta \rho^2}.$$

3. INTEGRAL FLOW / MULTICUT APPROXIMATION RATIOS

In this section, we apply the low-radius decompositions of Section 2 to prove our approximate max-integral-flow / min-multicut theorems. In fact, we show that there exist efficient algorithms which, given a lower bound on the weight of a min-multicut, either construct an integral flow achieving the claimed approximation ratio, or output a proof that the promise is violated in the form of a multicut with weight less than the asserted lower bound.

3.1 Proof of Theorem 1

Suppose we have an efficiently computable upper bound $f^*_{\mathcal{G}}(\rho) \geq f_{\mathcal{G}}(\rho)$ on f (from this point, we omit the subscript $\mathcal{G}$). Without loss of generality, we may assume f^* is monotonically decreasing in ρ. For all $\epsilon > 0$, define

$$g(\epsilon) = \min_{f^*(\rho) < \epsilon} 2\rho$$

where $g(\epsilon)$ is defined as ∞ if no such ρ exists. Note that $g(\epsilon)$ is also monotonically decreasing in ϵ and that g can be efficiently constructed using $O(n)$ invocations of f^*.

Theorem 1 is proved using the approximate integral flow algorithm illustrated on the facing page.

We claim that if the weight of a minimum K-cut of G is ϵC, then the algorithm produces a flow of weight F^* on input ϵ. Assuming this for a moment, it is immediate that the max-integral-flow / min-multicut ratio is bounded by $\epsilon C / F^*$ and so, to complete the proof of Theorem 1, we need only compute estimates of F^* given the functions $f^*_{\mathcal{G}^*}, f^*_{\mathcal{G}^{r,r}}, f^*_{\mathcal{G}^\delta}$ from Section 2. The following lemma states appropriate estimates. The computations are omitted from this abstract.

LEMMA 8. *Let $F^*_{\mathcal{G}}(\epsilon)$ denote the value of F^* computed using function $f^*_{\mathcal{G}}$ for input density ϵ. Then*

(i) $F^*_{\mathcal{G}^*}(\epsilon) \geq \frac{\epsilon^2}{16 \ln(k+1)} C$

(ii) $F^*_{\mathcal{G}^{r,r}}(\epsilon) \geq \frac{\epsilon^2}{16 r^2} C$

(iii) $F^*_{\mathcal{G}^\delta}(\epsilon) \geq \frac{\epsilon^{3/2} \delta^{1/2}}{48\sqrt{2}} C$

The asserted approximation ratios follow immediately from Lemma 8. We now prove the claim. Because the inner loop of the algorithm is an augmenting path process, the algorithm clearly produces an integral flow of weight t^*, where t^* is the last value of t reached in the main loop. Therefore, it is enough to show that, when the algorithm rejects, the algorithm outputs a K-cut of weight less than ϵC.

We begin by showing that W is a K-cut; that is, W intersects every K-path. There are two cases. First, we observe that every K-path in c_{t^*} intersects m. By the loop condition, there is no K-path in c_{t^*} of length at most $g(\epsilon_{t^*})$. Let $(S_1, \ldots, S_t)$ be any K-partitioning of V with c_{t^*}-radius $\leq \frac{1}{2} g(\epsilon_{t^*})$. If p is a K-path in c_{t^*} which does not intersect $\nabla_{c_{t^*}}(S_1, \ldots, S_t)$ then the entire path must lie in the same partition S_i. But

$$\mathrm{diam}_{c_{t^*}}(S_i) \leq 2\rho_{c_{t^*}}(S_i) \leq g(\epsilon_{t^*}).$$

Contradiction. Further, each edge $e \in m$ is also in W since $m(e) = c_{t^*}(e)$ implies that

$$[m + v_{t^*}](e) = [c_{t^*} + v_{t^*}](e) = c(e).$$

Second, we consider a K-path p such that p intersects an edge $e \in c$ but $\notin c_{t^*}$. Then

$$c(e) = c_{t^*}(e) + v_{t^*}(e) = v_{t^*}(e),$$

so $e \in W$ and p intersects W.

We next need to bound $c(W)$. Clearly,

$$c(W) \leq \|m\| + \|v_{t^*}\|.$$

When $t^* < F^*$, $\epsilon_{t^*} > 0$ and, since $\|p_i\| \leq g(\epsilon_i)$ for all i,

$$\|v_{t^*}\| = \sum_{j=0,\ldots,t^*-1} \|p_i\| \leq \sum_{j=0,\ldots,t^*-1} g(\epsilon_j) = (\epsilon - \epsilon_{t^*})C.$$

By definition of f^* and g, the K-cut $\nabla_{c_{t^*}}(S_1, \ldots, S_t)$ giving m exists (and, as in our case when the bounds on f^* are algorithmic, can be efficiently computed) and has

$$\begin{aligned} \|m\| = c_{t^*}(\nabla(S_1, \ldots, S_t)) &\leq f^*\left(\frac{1}{2} g(\epsilon_{t^*})\right) \|c_{t^*}\| \\ &< \epsilon_{t^*} \|c_{t^*}\| \leq \epsilon_{t^*} C. \end{aligned}$$

Summing these, $c(W) < (\epsilon - \epsilon_{t^*})C + \epsilon_{t^*} C = \epsilon C$.

To prove the claim concerning vertex covers in case (i), note that, as observed by Günlük [11] for the case of the fractional max-flow / min-multicut approximation ratio, it is sufficient to seed the partition selection step in the Garg-Vazirani-Yannakakis algorithm using a vertex cover K^* of K rather than the entire vertex set of K. The proof of the flow/cut approximation bound carries through in this case as well, since a K^*-partitioning is sufficient to intersect every sufficiently long shortest path between vertex-pairs in K. Thus, if k^* is the size of a minimum vertex cover of K, the approximation ratio is improved to $O(\epsilon^{-1} \log k^*)$. In an extreme case, for instance a star graph, $k = O(n)$ while setting K^* to the center vertex of the star gives $k^* = 1$.

It still remains to remove the assumption that we give the algorithm as input the weight ϵC of a min-multicut of G. Since $g(\epsilon_t)$ is an increasing function in t, we can remove the computation of the sequence $\{\epsilon_i\}$ altogether and replace the augmentation loop with a pure greedy algorithm; that is, while there exists a path between any two vertex-pairs of K in c_t, push one unit of flow along the *shortest* such path. This variant clearly produces at least as large a flow as the original algorithm on any input ϵ, in particular the

Approximate Max-Integral-Flow:

```
set ϵ_0 ← ϵ, v_0 ← (0, ..., 0), c_0 ← c
for t = 0, ..., ∞
  if ϵ_t ≤ 0 set F* ← t and break
  set ϵ_{t+1} ← ϵ_t − g(ϵ_t)/C
for t = 0, ..., ∞
  if ∃ k_j ∈ B_{c_t}(k_i, g(ϵ_t)) for some (k_i, k_j) ∈ K
    set p_t ← characteristic vector of any path of length ≤ g(ϵ_t) from k_i → k_j
    set v_{t+1} ← v_t + p_t, c_{t+1} ← c_t − p_t
  else break
if t ≥ F*
  output v_t and accept
else
  set m ← ∇_{c_t}(S_1, ..., S_l) s.t. ρ_{c_t}(S_i) ≤ ½ g(ϵ_t) and c_t(∇(S_1, ..., S_l)) < ϵ_t C
  output W = {e ∈ E | [v_t + m](e) = c(e), c(e) > 0} and reject
```

true value of ϵ. However, this version, as with the original, runs in time only weakly polynomial in the input size. This is easily corrected by modifying the greedy algorithm to push $\min_{e\in p} c_t(e)$ units of flow along the path p selected on each iteration. Then clearly there can be at most $|E|$ augmentation steps.

4. SOME APPLICATIONS

In this section, we observe that the proof of Theorem 1 yields efficient approximation algorithms for maximum integral and fractional multicommodity flow and related problems. We also observe some natural but less obvious combinatorial applications.

4.1 Approximation algorithms for flow and edge-disjoint path problems

In traditional applications of multicommodity max-flow / min-cut inequalities, the maximum flow problem is efficiently computable (via polynomial-time linear programming algorithms) while the corresponding cut problem is NP-hard. However, in the integral case, both the flow [8] and cut [5] problems are NP-hard. In the case where the min-multicut has weight at least ϵC, the approximate maximum integral flow algorithm used to prove Theorem 1 and the weak duality relations give the following.

Corollary 9. *Let G be a graph with $\mathbf{Z}^+$-valued capacity function c, and K be a demand graph on a k-element subset of V such that the weight of any multicut separating all pairs of vertices $(k_1, k_2) \in K$ is at least ϵC. Then*

(i) There is a polynomial-time algorithm which constructs an integral flow within a factor $O(\epsilon^{-1} \log k)$ of the optimum. If k^* is the vertex cover number of K, the ratio is improved to $O(\epsilon^{-1} \log k^*)$.

(ii) If G excludes $K_{r,r}$ for some constant r, the ratio is improved to $O(1/\epsilon)$.

(iii) If $c \in \{0,1\}^E$ and G is δ-dense, the ratio is improved to $O(1/\sqrt{\epsilon\delta})$.

A natural variation on the above is to apply scaling methods to recast the general (fractional) multicommodity flow problem as an integral flow problem. That is, fix some large integer Q and, for a non-negative *real*-valued capacity function c, let $c'(e) = \lfloor c(e)Q \rfloor$. If v' is an integral flow in c', then v'/Q is a feasible (possibly fractional) flow in c. Further, the scaling distorts the relative weight ϵ of the minimum cut and the original capacity function by a factor which goes to 1 as $Q \to \infty$.

Given a fractional capacity function c, consider the following *greedy heuristic* for the maximum multicommodity flow problem: Select a shortest K-path in G and push as much flow as possible along the path until it is saturated; repeat until there are no K-paths remaining.

Corollary 10. *Let G be a graph with a non-negative real-valued capacity function c, and let ϵ, k^* be as above. Then the greedy heuristic constructs a flow of weight within a factor $O(\epsilon^{-1} \log k^*)$ of the optimum. If G excludes $K_{r,r}$ for some constant r, then the factor is improved to $O(\epsilon^{-1})$.*

Proof. It is easy to see that the greedy heuristic corresponds to the pure greedy variant of the approximate max-integral-flow algorithm above when Q is an arbitrarily large integer. □

Note that we cannot apply scaling techniques to the family of δ-dense graphs. Unconditional versions of these approximation ratios were already known [7, 11, 13], but it is interesting that the greedy heuristic constructs a flow achieving these ratios when the min-multicut has constant density.

Integral flows are closely related to edge-disjoint paths in unweighted graphs. The problem of connecting a maximum number of endpoints in K along edge-disjoint paths was one of the original NP-hard problems [12]. The following consequence of Theorem 1 will be used later.

Corollary 11. *Let G be an unweighted graph with m edges and maximum degree Δ such that at least ϵm edges must be removed to separate all vertex-pairs in K. Then*

(i) $\Omega\left(\frac{\epsilon^2}{\Delta \log k} m\right)$ vertex-pairs in K can be connected along mutually edge-disjoint paths, and these paths can be computed in polynomial time.

(ii) If G excludes $K_{r,r}$ for some constant r, then the same is true for $\Omega\left(\frac{\epsilon^2}{\Delta} m\right)$ pairs.

(iii) If G is δ-dense, then the same is true for $\Omega\left(\frac{\epsilon^{3/2}\delta^{1/2}}{\Delta} m\right)$ pairs.

PROOF. Edge-disjoint paths in an unweighted graph are equivalent to the special case of integral flow where edge capacities are 0-1-valued. If G has maximum degree Δ, then at most Δ routed paths between a pair (k_1, k_2) in a K-flow can correspond to a single terminal pair. $\square$

4.2 Intersecting odd cycles

Let G be an unweighted graph, and let $og(G)$ denote the *odd girth* of G, that is, the length of the shortest odd cycle in G. In [3], Bollobás, Erdös, Simonovits, and Szemerédi considered the problem of determining the minimum cardinality of an edge-subset $F \subset E$ such that F intersects every odd cycle in G. They showed that $|F| = O\left(n^2/og(G)\right)$ and conjectured that $|F| = \Theta\left(n^2/og^2(G)\right)$. This conjecture was proved by Komlós [16] using the eponymous decomposition in Section 2.

Odd cycles in a graph have a very natural formulation in terms of integral multicommodity flows. The relation is summarized in the following lemma.

LEMMA 12. *Let $G(V, E)$ be an unweighted graph, and let (V_0, V_1) be a bipartition of V such that $|(V_0, V_0)| + |(V_1, V_1)|$ is minimized (that is, a maximum cut). Let $E = E_0 \cup E_1$ where $E_0 = (V_0, V_0) \cup (V_1, V_1)$ and $E_1 = (V_0, V_1)$, and set $K = E_0$. Then*

(i) Every K-path in $G_1 = G(V, E_1)$ can be extended by an edge in E_0 into an odd cycle in G.

(ii) For all $S \subset V$, $|\nabla_{G_1}(S)|/|\nabla_K(S)| \geq 1$.

(iii) Any K-cut of G_1 has size at least $|E(K)|$.

PROOF. Let p be a K-path in G_1, say, between $(k_1, k_2) \in E_0$. By definition, G_1 is bipartite, so p has even length. Then $p \cup (k_1, k_2)$ is an odd cycle in G. Next, suppose there exists $S \subset V$ such that $|\nabla_{G_1}(S)| < |\nabla_K(S)|$. Then setting

$$V_i' = V_i - (V_i \cap S) + (V_{\bar{i}} \cap S)$$

gives $|(V_0', V_0')| + |(V_1', V_1')| = |(V_0, V_0)| + |(V_1, V_1)| + |\nabla_{G_1}(S)| - |\nabla_K(S)| < |(V_0, V_0)| + |(V_1, V_1)|$, contradicting the minimality in the choice of (V_0, V_1). Finally, let M be a K-cut of G_1, and let $S_1, \ldots, S_t$ denote the connected components of $G_1 \setminus M$. Then $\sum_i \nabla_K(S_i) = 2|E(K)| \leq \sum_i \nabla_{G_1}(S_i) \leq 2|M|$. $\square$

THEOREM 13. *If $G \in \mathcal{G}$ then $|F| \leq f_{\mathcal{G}}(\frac{1}{2}(og(G) - 3))m$. In particular,*

(i) $|F| = O\left(m \log n/og(G)\right)$

(ii) $|F| = O\left(n^2/og^2(G)\right)$

(iii) If G excludes $K_{r,r}$ for some constant r, then $|F| = O\left(m/og(G)\right)$.

PROOF. By definition of $f_{\mathcal{G}}$, there exists a vertex-partition $(S_1, \ldots, S_t)$ with $|\nabla(S_1, \ldots, S_t)| \leq f_{\mathcal{G}}(\frac{1}{2}(og(G) - 3))m$ such that $\rho(S_i) < \frac{1}{2}(og(G) - 1)$. It is easy to see that such a multicut intersects every odd cycle of length at least $og(G)$. The remaining claims follow by applying the upper bounds f^* proved in Section 2. $\square$

Each of these bounds can also be interpreted as an upper bound on $og(G)$ in terms of $|F|$. Note that the first bound is stronger when G is a sparse graph; the second is Komlós' result; the third is a stronger bound for sparse graphs in the case of graphs excluding $K_{r,r}$.

4.3 Property testing

We note an application to *property testing*, a model of computation introduced by Goldreich, Goldwasser, and Ron [9, 10]. In one formulation of this model, an algorithm is given oracle access to the adjacency list of a graph G with constant maximum degree. The goal is to determine, using as small a number of queries as possible and with probability bounded away from $\frac{1}{2}$, whether the graph has a given property or is far from having the property in the sense that at least a constant fraction of the entries of the adjacency table must be modified to carry G into a graph with the property. One is generally interested in algorithms which make a number of queries which is *sub-linear* in the size of G (in fact, often *independent* of the size of G).

Consider the problem of testing whether G is bipartite. Suppose that we would like an algorithm which, given a graph which is ϵ-far from bipartite (at least ϵm edges must be deleted to make it bipartite), locates a witness to the non-bipartiteness of G (that is, an odd cycle). Such an algorithm obviously has one-sided error, and the probability of error is exactly the probability that it fails to find an odd cycle in a far-from-bipartite graph. The following lemma states that a graph which is ϵ-far from bipartite is dense in small witnesses to this fact.

LEMMA 14. *Let G be a graph of constant maximum degree which is ϵ-far from bipartite. Then G contains $\Omega(\epsilon^2 m/\log n)$ edge-disjoint odd cycles of length $O(\epsilon^{-2} \log n)$. If G excludes $K_{r,r}$ for some constant r, then it contains $\Omega(\epsilon^2 m)$ edge-disjoint odd cycles of length $O(\epsilon^{-2})$.*

PROOF. With notation as earlier, $|F| \geq \epsilon m$. Apply Corollary 11 to the construction in Lemma 12. That a constant fraction of these cycles have at most the given length follows by Markov's inequality. $\square$

Lemma 14 implies an efficient bipartiteness testing algorithm for graphs excluding $K_{r,r}$:

THEOREM 15. *Let G be a graph with constant maximum degree such that G excludes $K_{r,r}$ for some constant r and G is ϵ-far from bipartite. Then there exists an algorithm which locates an odd cycle in G with probability $1 - \delta$ using $\exp(O(\epsilon^{-2})) \log(1/\delta)$ queries; in particular, the algorithm requires a number of queries which is independent of n.*

PROOF. In the case of graphs excluding $K_{r,r}$, Lemma 14 implies that we can locate an odd cycle with constant probability by sampling $O(\epsilon^{-2})$ random vertices and doing a breadth-first search about each vertex to radius $O(\epsilon^{-2})$. Repeating this $O(\log(1/\delta))$ times reduces the failure probability to $1 - \delta$. The overall query complexity of this procedure is $\exp(O(\epsilon^{-2})) \log(1/\delta)$. $\square$

On the other hand, Goldreich and Ron [10] showed that locating an odd cycle in this model requires $\Omega(\sqrt{n})$ queries in general! A slightly modified argument shows that a graph with $|F| \geq \epsilon n^2$ must contain $\Omega(\epsilon^{3/2} n^2)$ edge-disjoint odd cycles. Alternatively, one can argue directly from Lemma 4 that bipartiteness as well as other properties on bounded-degree graphs, including 3-colorability (which is not testable with a sub-linear number of queries [2]), are testable in constant time on graphs excluding a fixed minor.

5. OTHER COMMENTS

With respect to the usual parameter k, the bounds of Theorem 1 are optimal. For general graphs, this follows from the lower bound in [7], which is similar to the lower bound construction used in [17]: G is a Δ-regular expander graph on n vertices, all edges have unit capacity, and K is the set of all vertex-pairs (k_1, k_2) such that $d(k_1, k_2) \geq \log_\Delta(n/2)$. It is easy to check that the maximum (fractional) flow is $O(n/\log n)$ while the minimum multicut has weight $\Omega(n) = \Omega(m)$. Fractional flow is a relaxation of integral flow, so the bound holds in our case as well, even though the min-multicut has constant density. For the other families we considered, the assertion is trivial.

It is not clear, however, whether Theorem 1 always captures correctly the optimal dependence of the max-integral-flow / min-multicut ratio on ϵ. For planar graphs, the grid example used in [8] to establish the $\Omega(k)$ integrality gap shows that our dependence on ϵ for planar graphs is optimal. More generally, a potential problem in our approach is illustrated by the following example, which is the construction of Lemma 12 applied to the lower bound construction used in [3]. Let G' be a path on $l = O(1/\sqrt{\beta})$ vertices $v'_1, \ldots, v'_l$ and let G be the "blow-up" of G' by $t = O(\sqrt{\beta}n)$ vertices; that is, replace each vertex v'_i of G' by t vertices v_i^j for $j = 1, \ldots, t$, and let G be the graph formed by connecting all pairs $(v_j^{i_1}, v_{j+1}^{i_2})$ for all i_1, i_2, j. Set K to the set of pairs $(v_1^{i_1}, v_l^{i_2})$ for all i_1, i_2. Then G is δ-dense, with $\delta = \sqrt{\beta}$, and it is easy to see that any K-cut has weight $\Omega(\sqrt{\beta}C)$. Applying Theorem 1 with $\epsilon = \sqrt{\beta}$ gives that there exists a flow of weight $O(\beta C)$ when, in fact, there exists a flow with weight equal to the min-multicut – the optimum flow is achieved by pushing a unit flow along all paths $\mathcal{P} = \{(a_1, \ldots, a_l)\}$ where $\mathcal{P}$ is a pairwise independent family of vectors on $\{1, \ldots, t\}^l$ of size t^2 (this flow strategy was observed by Luca Trevisan). The problem is that, when we augment along a path, it may be the case that the path intersects a min-multicut at a single edge, while the approximate max-integral-flow algorithm reduces its lower bound on the weight of the unknown min-multicut by the length of the path. In this example, the worst case occurs on every augmentation, leading to a loss of a factor $l = O(1/\sqrt{\beta})$. On the other hand, the same example shows that our bound on f_{G^δ} in this case is optimal within a constant factor.

6. ACKNOWLEDGEMENTS

We thank Andrej Bogdanov, Chris Harrelson, Satish Rao, and Luca Trevisan for helpful comments and discussions.

7. REFERENCES

[1] Y. Aumann and Y. Rabani. An $O(\log k)$ approximate min-cut max-flow theorem and approximation algorithm. *SIAM Journal on Computing*, 27(1):291–301, 1998.

[2] A. Bogdanov, K. Obata, and L. Trevisan. A lower bound for testing 3-colorability. In *IEEE Symposium on Foundations of Computer Science*, pages 93–102, 2002.

[3] B. Bollobás, P. Erdös, M. Simonovits, and E. Szemerédi. Extremal graphs without large forbidden subgraphs. *Annals of Discrete Mathematics*, 3:29–41, 1978.

[4] C. Chekuri and S. Khanna. Edge disjoint paths revisited. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 628–637, 2003.

[5] E. Dahlhaus, D. Johnson, C. Papadimitriou, P. Seymour, and M. Yannakakis. The complexity of multiterminal cuts. *SIAM Journal on Computing*, 23:864–894, 1994.

[6] L. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8, 1956.

[7] N. Garg, V. V. Vazirani, and M. Yannakakis. Approximate max-flow min-(multi)cut theorems and their applications. In *ACM Symposium on Theory of Computing*, pages 698–707, 1993.

[8] N. Garg, V. V. Vazirani, and M. Yannakakis. Primal-dual approximation algorithms for integral flow and multicut in trees. *Algorithmica*, 18:3–20, 1997.

[9] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, 45(4):653–750, 1998.

[10] O. Goldreich and D. Ron. Property testing in bounded degree graphs. In *ACM Symposium on Theory of Computing*, pages 406–415, 1997.

[11] O. Günlük. A new min-cut max-flow ratio for multicommodity flows. In *Integer Programming and Combinatorial Optimization*, pages 54–66, 2002.

[12] R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103, New York, NY, 1972. Plenum Press.

[13] P. Klein, S. Plotkin, and S. Rao. Excluded minors, network decomposition, and multicommodity flow. In *ACM Symposium on Theory of Computing*, pages 682–690, 1993.

[14] J. Kleinberg. *Approximation algorithms for disjoint paths problems*. Ph.D. Thesis, MIT, Cambridge, MA, 1996.

[15] S. G. Kolliopoulos and C. Stein. Approximating disjoint-path problems using greedy algorithms and packing integer programs. In *Integer Programming and Combinatorial Optimization*, 1998.

[16] J. Komlós. Covering odd cycles. *Combinatorica*, pages 393–400, 1997.

[17] F. T. Leighton and S. Rao. An approximate max-flow min-cut theorem for uniform multi-commodity flow problems with applications to approximation algorithms. *Journal of the ACM*, 46(6), 1999.

[18] N. Linial, E. London, and Y. Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15:215–245, 1995.

[19] S. Plotkin and E. Tardos. Improved bounds on the max-flow min-cut ratio for multicommodity flows. In *ACM Symposium on Theory of Computing*, pages 691–697, 1993.

[20] K. Varadarajan and G. Venkataraman. Graph decomposition and a greedy algorithm for edge-disjoint paths. In *ACM-SIAM Symposium on Discrete Algorithms*, 2004.