A Note on the Greedy Algorithm for the Unsplittable Flow Problem

Petr Kolman
Inst. for Theoretical Computer Science*
Charles University
Malostranské nám. 25
118 00 Prague, Czech Republic
kolman@kam.mff.cuni.cz

November 22, 2002

Abstract

In a recent paper Chekuri and Khanna improved the analysis of the Greedy algorithm for the Edge Disjoint Paths problem and proved the same bounds also for the related Uniform Capacity Unsplittable Flow Problem. Here we show that their ideas can be used to get the same approximation ratio even for the more general Unsplittable Flow Problem with nonuniform edge capacities.

*Supported by the Ministry of Education of the Czech Republic as project LN00A056.

1 Introduction

In the maximum *edge disjoint paths* problem (EDP) we are given a (directed on undirected) graph $G = (V, E)$ and a set $T = \{(s_i, t_i) : 1 \le i \le k\}$ of k requests. The objective is to connect a maximum number of pairs from T along edge disjoint paths. The *unsplittable flow problem* (UFP) is a natural generalization: the edges in G have capacities prescribed by $c : E \to \mathbb{R}_+$ and every request r_i in T has an associated demand d_i. A feasible solution is a subset of requests from T such that all of them can be routed in G without violating the capacity constraints, each request along a single path. The objective is to find a feasible solution with maximal total demand. Throughout the paper we will assume that the maximal demand and the maximal edge capacity is at most one. This is not a restriction since it can be always achieved by a suitable scaling of the demands and the capacities.

A common assumption in dealing with the UFP is that the minimal edge capacity is one, that is, the minimal capacity is at least as large as the maximal demand. We will call this a *no bottleneck assumption*. A *unit capacity* unsplittable flow problem (UCUFP) is a variant of the problem in which all capacities are equal to one.

For the UCUFP Kleinberg [5] proved an $O(\sqrt{m} \cdot \frac{\max c(e)}{\min d_i})$ approximation ratio of a greedy algorithm, where m is the number of edges in G. The upper bound was improved by Bajeva and Srinivasan [2] to $O(\sqrt{m})$ for the slightly more general UFP under the no-bottleneck assumption, by an algorithm based on randomized rounding of an optimal fractional solution of the corresponding LP. Later on, Kolman and Scheideler [7] presented a variant of the greedy algorithm with an $O(\sqrt{m})$ approximation ratio even for the UFP *without* the no-bottleneck assumption.

For the simpler EDP, the best approximation ratio pending for a long time was also only $O(\sqrt{m})$. In a recent work, Chekuri and Khanna [3] substantially extended the understanding of the problem by improving the analysis of the greedy algorithm for dense graphs. They proved an $O(\min\{n^{2/3}, \sqrt{m}\})$ upper bound for undirected graphs and $O(\min\{n^{4/5}, \sqrt{m}\})$ for directed graphs. Moreover, they showed that the same bounds apply also for the unit capacity UFP: the point was that requests with demand more than $1/2$ behave like requests in the EDP, and for requests with demand at most $1/2$ there already was an algorithm with $O(\sqrt{n})$ approximation ratio [2, 1, 7].

We pause to note that for many graph classes the approximation ratio of the greedy algorithm is much better (e.g., $O(\log n)$ for expander or hypercubic graphs; $O(F)$ in general, where F is the flow number of the respective graph [7]), however, in terms of n this is only $O(n)$ in the worst case of graphs with $\Omega(n^2)$ edges.

On the lower bound side, in terms of the number of edges m, it was shown by Guruswami et al. [4] that on directed networks it is NP-hard to approximate the UFP within a factor of $m^{1/2-\epsilon}$, for any $\epsilon > 0$. In terms of the number of vertices n, the above lower bound is $n^{1/2-\epsilon}$.

The contribution of this paper is that the same bounds as for the EDP can be achieved for the UFP on graphs with non-uniform edge capacities and even for the UFP without the no-bottleneck assumption. The presented algorithm combines the ideas from the previous best algorithm for the UFP without the no-bottleneck assumption [7] and the above mentioned improved algorithm for the EDP [3]. Its advantage is also that it is completely selfcontained and does not use other algorithms as subroutines.

The main open problem is the same as for the EDP: is it possible to approximate the UFP within $O(\sqrt{n})$ in polynomial time?

2 Tools

The basic idea of the $\sqrt{m}$ approximation algorithm for the EDP is the simple fact that there can be in G at most $\sqrt{m}$ edge disjoint paths of length $\sqrt{m}$ and more. Chekuri and Khanna observed that even a more strict restriction holds:

Theorem 2.1 (Chekuri and Khanna, 2003) *Let $G(V, E)$ be a simple unit capacity undirected graph and let T be the collection of all source-sink pairs such that the shortest distance in G between source and sink is at least l. Then the value of the maximum multicommodity flow for pairs in T is bounded by $O(n^2/l^2)$.*

A slightly weaker but similar result holds for directed graphs too.

We will need analogous bounds for capacitied networks and requests with demands. The presented proofs of these bounds follow the lines of the original proofs for the EDP. However, in our case, the situation is complicated since many paths may use a single edge while some edges are unavailable for some requests due to small capacity. More care must be given to the *distances* and *degrees* in the network. It is worth noting that in contrast to the original proof, here we essentially use the fact that the paths in the optimal UFP solution are *unsplittable*. Most of the previous approximations for the EDP and the UFP are based on comparison with an optimal *fractional* solution [2, 7, 3].

Let $G(V,E)$ be a (directed or undirected) network with edge capacities prescribed by $c : E \to \mathbb{R}_+$. A *d-path* is a path carrying d units of flow and we talk about an $s-t$ path if we want to stress that the terminal vertices are s and t. An $s-t$ d-path is *feasible* if the path obeys the capacity constraints. An edge e is *d-critical* if $c(e) \geq d > c(e)/2$. For a source-sink pair (s,t) with demand d, let the *demand-d-distance* $\mathrm{dist}_d(s,t)$ from s to t be the minimal number of d-critical edges on a feasible $s-t$ d-path. If there is no such path, the distance is infinite.

Lemma 2.2 *Let $G(V,E)$ be an undirected network with edge capacities and let $d_{\max} \in (0,1]$ and $l \in \mathbb{N}$. For every pair of nodes (s_i,t_i) define a demand d_i such that $dist_{d_i}(s,t)$ is at least l and d_i is the minimal possible. Let T be the collection of all source-sink pairs (s_i,t_i) with demand $d_i \leq d_{\max}$ and let $\mathcal{O}$ be the optimal solution for the UFP instance T. Then the total flow in $\mathcal{O}$ is bounded by $O(d_{\max} n^2/l^2)$.*

Proof. We say that a vertex u in G is a *high degree vertex with respect to demand d* if the number of d-critical edges adjacent to u is more than $6n/l$, and is a *low degree vertex with respect to d* otherwise. If it is clear from the context with respect to what demand are we talking about the degree, we will talk about high and low degree vertices only.

Claim 2.3 *For every request $(s,t) \in \mathcal{O}$ with demand d, the $s-t$ d-path in $\mathcal{O}$ uses at least $l/6$ d-critical edges adjacent to low degree vertices.*

Proof. Suppose by contradiction that there exists a request $(s,t) \in \mathcal{O}$ with an $s-t$ d-path with less than $l/6$ d-critical edges adjacent to low degree vertices. Let G_d be the subgraph of G containing only edges with capacity at least d and let G'_d denote the graph we get when contracting pairs of vertices connected in G_d by an edge with capacity $2d$ and more. Note that the number of edges on any $s-t$ path is still at least l and that there are only d-critical edges in G'_d.

Now it is possible to use the arguments of Chekuri and Khanna. Let τ be a breadth-first search tree of G'_d rooted at s and let L_i (*layer*) denote the set of nodes at distance i from s in τ. Assume that among the first l layers, there are at most $l/6-1$ layers consisting entirely of low degree vertices (otherwise the claim is obvious). Let a block B_i be a set of nodes on layers $L_{3i+1}, L_{3i+2}, L_{3i+3}$, $0 \leq i < l/3$. By our assumption there are at least $l/6$

blocks containing no layer consisting entirely of low degree vertices. Since there are at most n vertices, one of these blocks, say a block B_j, contains at most $6n/l$ nodes. However, this is a contradiction since all nodes in the middle level of B_j, in the level $3j+2$, must be low degree nodes - they are adjacent to nodes in B_j only.

Since the number of d-critical edges adjacent to low degree vertices on a shortest $s-t$ d-path is the same in G and in G'_d, the proof is completed. □

Consider an interval $(1/2^{k+1}, 1/2^k]$, for some integral k. By the above claim, every request in $\mathcal{O}$ with demand $d \in (1/2^{k+1}, 1/2^k]$ uses at least $l/6$ $(1/2^{k+1})$-critical and $(1/2^k)$-critical edges adjacent to low degree vertices (with respect to $1/2^{k+1}$ and $1/2^k$, resp.). Since each vertex of a low degree with respect to d has at most $6n/l$ d-critical edges, there are only $O(n^2/l^2)$ requests with demand $d \in (1/2^{k+1}, 1/2^k]$ in the optimal solution. Since the maximal single demand in $\mathcal{O}$ is $d_{\max}$, the total flow in $\mathcal{O}$ is bounded by $O(d_{\max} n^2/l^2)$. □

An analogous lemma will be needed for directed networks:

Lemma 2.4 *Let $G(V, E)$ be a directed network with edge capacities and let $d_{\max} \in (0, 1]$ and $l \in \mathbb{N}$. For every pair of nodes (s_i, t_i) define a demand d_i such that $dist_{d_i}(s, t)$ is at least l and d_i is the minimal possible. Let T be the collection of all source-sink pairs (s_i, t_i) with demand $d_i \le d_{\max}$ and let $\mathcal{O}$ be the optimal solution for the UFP instance T. Then the total flow in $\mathcal{O}$ is bounded by $O(d_{\max} n^4/l^4)$.*

Proof. The goal is to find a cut $E' \subseteq E$ with total capacity $O(d_{\max} n^4/l^4)$ separating all pairs in T. The cut E' is constructed iteratively. Set $E' = 0$ at the beginning. Let $G' = G(V, E - E')$ and choose a request $(s, t) \in T$ with the minimal demand, say d, that still has a feasible path in G'. Remove from G' all edges with capacity strictly less than d and contract all pairs of nodes connected by an edge with capacity $2d$ and more. Let G'_d denote the resulting network, τ the breadth-first search tree of G'_d rooted at s and L_i (*layer*) the set of nodes at distance i from s in τ.

Claim 2.5 *There is a j, $l/3 < j \le 2l/3$, such that the number of edges going in G'_d from L_j to L_{j+1} is $O(n^2/l^2)$. Moreover, these edges separate $\Omega(l^2)$ pairs of nodes that have been connected in G'_d.*

Proof. Apply the argument of Chekuri and Khanna: let $B_i = L_{2i} \cup L_{2i+1}$, $i \ge 0$. By the construction of T, we have $t \in B_r$ for some $r \ge l/2$. Since there are at most n nodes, there must be a B_k with $l/6 \leq k < l/3$ and size at most $6n/l$. Clearly, the number of edges from L_{2k} to L_{2k+1} is $O(n^2/l^2)$. Moreover, by the construction of τ, these edges separate the first $l/6$ nodes on an $s-t$ path from the last $l/6$ nodes on the same path, in total $\Omega(l^2)$ pairs. □

Since we take as (s,t) the minimal request with a feasible path in G', the cut from the above construction applied to G' (it was defined for G'_d) separates $\Omega(l^2)$ pairs than have been connected in G' by a path with flow d or more at the beginning of the iteration. Thus, after $O(n^2/l^2)$ iterations all pairs from T are separated and in each iteration the size of the cut is $O(d_{\max}n^2/l^2)$. Since the size of the cut is an upper bound on he size of an UFP solution, the proof is completed. □

3 The Bounded Greedy Algorithm

There are several different versions of the Greedy algorithm. Chekuri and Khanna use the following one for the EDP (and, actually, it is also one of the basic building blocks of their UFP algorithm): for each unrouted request compute its shortest feasible path and connect the pair with the minimum shortest path (cf. [6]). Kleinberg [5] proposed for the EDP the Bounded greedy algorithm (BGA) and used it with parameter $l = \sqrt{m}$: process the requests in any order and if there is a free path of length at most l connecting the terminal nodes, accept the request and use any such path for it.

We will use for the UFP with non-uniform edge capacities an extended variant of the BGA called careful BGA [7]: process the requests according to their demands, starting with the largest. Accept a request r if there exists a feasible path p for it such that after routing r the total flow on at most l edges of p is larger than half of their capacity. We say that the request r uses these edges in their *upper half.* Let $\mathcal{B}_1$ denote the solution we get. Let $\mathcal{B}_2$ denote the solution consisting of only the request with the maximal demand $d_{\max}$ (without loss of generality we assume that such a request always fits in the empty network). As our solution we take the better of these two, $\mathcal{B} = \max(\mathcal{B}_1, \mathcal{B}_2)$.

For a set $\mathcal{U}$ of flow paths let $||\mathcal{U}||$ denote the total flow through all of then.

Theorem 3.1 *The approximation ratio of the careful BGA with parame-*

ter $l = \min\{\sqrt{m}, n^{2/3}\}$ *on undirected networks, and with parameter* $l = \min\{\sqrt{m}, n^{4/5}\}$ *on directed networks, is* $O(l)$ *for the UFP, even without the no-bottleneck assumption.*

Proof. Since the edge capacities may substantially vary, the separation of the requests into those with demand at most 1/2 and those with demand more than 1/2 [3] does not help in this setting and a different approach must be taken. We will assume that for undirected networks $l = n^{2/3}$ and for directed networks $l = n^{4/5}$ since for $l = \sqrt{m}$ the result is already known [7].

Let $\mathcal{O}$ denote the optimal solution and $\mathcal{O}' \subseteq \mathcal{O}$ its subset consisting of requests rejected by the first run of the careful BGA algorithm, that is of requests not in $\mathcal{B}_1$. Obviously, $||\mathcal{O} - \mathcal{O}'|| \leq ||\mathcal{B}||$. Consider a path $p \in \mathcal{O}'$. There are two possible reasons why the request r corresponding to p is not in $\mathcal{B}_1$: either (1) p was infeasible, which means the existence of an edge $e \in p$ where r did not fit in, or (2) there are (at least) l edges $e_1, \cdots, e_l$ on p that would be used by r in their upper half, that is for each e_i the sum of $d(p)$ and the flow on e_i in the moment of deciding about p was larger than half of their capacity $c(e_i)/2$. Let $\mathcal{O}_1 \subseteq \mathcal{O}'$ denote the requests rejected by the BGA for the first reason and $\mathcal{O}_2 \subseteq \mathcal{O}'$ for the second, $\mathcal{O}_2 = \mathcal{O}' - \mathcal{O}_1$.

Consider a request $p \in \mathcal{O}_1$. Since the requests were processed according to their demands, the flow on some $e \in p$ in the moment of rejecting p was strictly more than $c(e)/2$. Let $\mathcal{B}_e$ denote the paths from $\mathcal{B}_1$ participating on this flow that use the edge e in the upper half. Due to processing the requests according to their demands, the sum of flows of paths in $\mathcal{B}_e$ is at least $c(e)/4$: if $d(p) \leq c(e)/4$, it follows from the lack of space for p on e, and if $d(p) > c(e)/4$, there is a path q using e in the upper half with $d(q) \geq d(p)$. Each of paths $q \in \mathcal{B}_e$ is a *witness* for p and its weight for p is $d(q) \cdot d(p)/c(e)$. Note that the total weight of each path $q \in \mathcal{B}_1$ as a witness for paths in $\mathcal{O}_1$ is at most $d(q) \cdot l$, and, on the other hand, each path $p \in \mathcal{O}_1$ has witnesses in $\mathcal{B}_1$ with total weight at least $d(p)/4$. Thus $||\mathcal{O}_1|| \leq 4l \cdot ||\mathcal{B}_1||$.

We further distinguish reasons for rejecting $p \in \mathcal{O}_2$: either (a) there are more than $l/2$ edges on p such that $d(p) > c(e)/2$ for each of them, or (b) there are $l/2$ edges $e \in p$ each with a flow at least $c(e) - d(p) \geq d(p)$. Let $\mathcal{O}_2'$ be the subset of $\mathcal{O}_2$ consisting of requests rejected for the reason (a) and let G' be the network G with the capacity of each edge decreased by the flow along it in $\mathcal{B}_1$. Since for undirected (directed, resp.) networks, $\mathcal{O}_2'$ is a subset of T from Lemma 2.2 (Lemma 2.4, resp.), $||\mathcal{O}_2'|| = O(d_{\max} n^2/l^2) = O(l \cdot ||\mathcal{B}||)$ for $l = n^{2/3}$ ($||\mathcal{O}_2'|| = O(d_{\max} n^4/l^4) = O(l \cdot ||\mathcal{B}||)$ for $l = n^{4/5}$, resp.).

Consider now a path $p \in \mathcal{O}_2 - \mathcal{O}'_2$. The reason for rejecting p guarantees that for $d(p)$ units of flow along p in the optimal solution, we have $l \cdot d(p)/2$ units of *volume* in $\mathcal{B}_1$, and any request $r \in \mathcal{B}_1$ contributes to this volume by at most $l \cdot d(r)$ units. Thus, $||\mathcal{O}_2 - \mathcal{O}'_2|| = O(||\mathcal{B}_1||)$.

Combining bounds on the partitions of $\mathcal{O}$ completes the proof. □

References

[1] Y. Azar and O. Regev. Strongly polynomial algorithms for the unsplittable flow problem. In *Proceedings of the 8th Conference on Integer Programming and Combinatorial Optimization (IPCO)*, 2001.

[2] A. Baveja and A. Srinivasan. Approximation algorithms for disjoint paths and related routing and packing problems. *MOR: Mathematics of Operations Research*, 25, 2000.

[3] C. Chekuri and S. Khanna. Edge disjoint paths revisited. In *Proceedings of the 14th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2003.

[4] V. Guruswami, S. Khanna, R. Rajaraman, B. Shepherd, and M. Yannakakis. Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems. In *Proceedings of the 31st ACM Symposium on Theory of Computing (STOC)*, pages 19–28, 1999.

[5] J. Kleinberg. *Approximation Algorithms for Disjoint Paths Problems.* PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1996.

[6] S. G. Kolliopoulos and C. Stein. Approximating disjoint-path problems using greedy algorithms and packing integer programs. In *Proceedings of the 6th Integer Programming and Combinatorial Optimization Conference (IPCO)*, volume 1412 of *Lecture Notes in Computer Science*, pages 153–162, 1998.

[7] P. Kolman and C. Scheideler. Improved bounds for the unsplittable flow problem. In *Proceedings of the 13th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 184–193, 2002.