Noname manuscript No.
(will be inserted by the editor)

The All-or-Nothing Flow Problem in Directed Graphs with Symmetric Demand Pairs

Chandra Chekuri · Alina Ene

Received: date / Accepted: date

Abstract We study the approximability of the All-or-Nothing multicommodity flow problem in directed graphs with symmetric demand pairs (SymANF). The input consists of a directed graph $G = (V, E)$ and a collection of (unordered) pairs of nodes $\mathcal{M} = \{s_1t_1, s_2t_2, \ldots, s_kt_k\}$. A subset $\mathcal{M}'$ of the pairs is *routable* if there is a feasible multicommodity flow in G such that, for each pair $s_it_i \in \mathcal{M}'$, the amount of flow from s_i to t_i is at least one *and* the amount of flow from t_i to s_i is at least one. The goal is to find a maximum cardinality subset of the given pairs that can be routed. Our main result is a poly-logarithmic approximation with constant congestion for SymANF. We obtain this result by extending the well-linked decomposition framework of [9] to the directed graph setting with symmetric demand pairs. We point out the importance of studying routing problems in this setting and the relevance of our result to future work.

Supported in part by NSF grants CCF-1016684 and CCF-1319376. Part of this work was done while the author was supported by TTI Chicago on a sabbatical visit in Fall 2012.

Supported in part by NSF grants CCF-1016684 and CCF-0844872. Part of this work was done while the author was an intern at TTI Chicago.

Chandra Chekuri
Department of Computer Science, University of Illinois at Urbana-Champaign
E-mail: chekuri@illinois.edu

Alina Ene
Center for Computational Intractability, Princeton University
Department of Computer Science and DIMAP, University of Warwick
E-mail: aene@cs.princeton.edu

1 Introduction

We consider some fundamental maximum throughput routing problems in *directed* graphs. In this setting, we are given a capacitated directed graph $G = (V, E)$ with n nodes and m edges. We are also given source-destination pairs of nodes $(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)$. The goal is to select a largest subset of the pairs that are simultaneously *routable* subject to the capacities; a set of pairs is routable if there is a multicommodity flow for the pairs satisfying certain constraints that vary from problem to problem (e.g., integrality, unsplittability, edge or node capacities). Two well-studied optimization problems in this context are the Maximum Edge Disjoint Paths (MEDP) and the All-or-Nothing Flow (ANF) problem. In MEDP, a set of pairs is routable if the pairs can be connected using edge-disjoint paths. In ANF, a set of pairs is routable if there is a feasible multicommodity flow that fractionally routes one unit of flow from s_i to t_i for each routed pair (s_i, t_i). ANF, introduced in [12,8], can be seen as a relaxed version of MEDP where the flow for the routed pairs is not required to be integral.

MEDP and ANF are both **NP**-hard and their approximability has attracted substantial attention. Over the last decade, several non-trivial results on both upper bounds and lower bounds have led to a much better understanding of these problems. At a high level, one can summarize this progress as follows. MEDP and ANF admit poly-logarithmic approximation in *undirected* graphs if one allows constant congestion[1]; in fact, a congestion of 2 is sufficient for MEDP [16] and for ANF no extra congestion is needed [8]. Moreover, both problems are hard to approximate to within a factor of $\Omega(\log^{\frac{1-\varepsilon}{c+1}} n)$ for any constant congestion $c \geq 1$ [2]; the hardness is under the assumption that $\mathbf{NP} \not\subseteq \mathbf{ZPTIME}(n^{\text{polylog}(n)})$. In sharp contrast, in *directed* graphs both problems are hard to approximate to within a *polynomial* factor for any constant congestion $c \geq 1$; the hardness factor is $n^{\Omega(1/c)}$ [14]. The upper bounds and lower bounds on the approximability are closely related to corresponding integrality gap bounds on a multicommodity flow relaxation for these problems.

In this paper, with several interrelated motivations in mind that we discuss in detail subsequently, we initiate the study of maximum throughput routing problems in directed graphs in the setting where the demand pairs are *symmetric*. Informally, in a symmetric demand pair instance, the input pairs are *unordered* and a pair $s_i t_i$ is routed only if both the ordered pairs (s_i, t_i) and (t_i, s_i) are routed. In particular, we focus our attention on the SymANF problem. The input consists of a directed graph $G = (V, E)$ and a collection of (unordered) pairs of nodes $\mathcal{M} = \{s_1 t_1, s_2 t_2, \ldots, s_k t_k\}$. A subset $\mathcal{M}'$ of the pairs is *routable* if there is a feasible multicommodity flow in G such that, for each pair $s_i t_i \in \mathcal{M}'$, the amount of flow from s_i to t_i is one unit *and* the

[1] A routing has congestion c if it violates the capacities by a factor of at most c.

amount of flow from t_i to s_i is one unit[2]. The goal is to find a maximum cardinality subset of the given pairs that can be routed.

One issue is whether we assume capacities on edges or on nodes or on both. In undirected graphs the node capacitated case is more general, however, in directed graphs this is not the case. In this paper we will assume that G has only node-capacities, in particular that each node has capacity one. The main reason for our choice is to relate routability in the symmetric setting to the notion of directed treewidth.

Our main result is the following theorem that gives a poly-logarithmic approximation with constant congestion for SymANF.

Theorem 1 *There is a polynomial time algorithm that, given any instance of the* SymANF *problem in directed graphs, it routes* $\Omega(\mathrm{OPT}/\log^2 k)$ *pairs with constant node congestion, where* OPT *is the value of an optimal fractional solution for the instance.*

The congestion that we guarantee is 64. We believe that the congestion can be improved, but we have not attempted to optimize the constant. Our algorithm uses a natural LP relaxation for the problem as a starting point and we also show a poly-logarithmic upper bound on the integrality gap of the relaxation. Some simple and natural extensions such as handling capacitated graphs and pairs with demand values can be handled via known techniques and we do not address them in this version.

We observe that, via existing results on the hardness of ANF in undirected graphs with congestion [2], one can conclude that SymANF with congestion c is hard to approximate to within a factor of $(\log n)^{\Omega(1/c)}$ for any fixed c unless $\mathbf{NP} \subseteq \mathbf{ZPTIME}\left(n^{\mathrm{polylog}(n)}\right)$.

2 Motivation and connection to related problems

The study of routing problems is motivated by several real-world applications but also by the fundamental role that flows and cuts play in algorithms, combinatorial optimization, and graph theory. For a single pair (s, t) it is well-known that the value of a maximum s-t flow in a directed graph is equal to the value of a minimum s-t cut; moreover, when the capacities are integral, the maximum fractional s-t flow is equal to the maximum integral s-t flow. These nice structural properties do not hold in the multicommodity setting in undirected or directed graphs even when the number of commodities is three.

The study of *approximate* flow-cut gap results, starting with the seminal work of Leighton and Rao [27], has been extremely fruitful and we now have

[2] There are alternative ways to define routability that captures symmetry. One option is to require a flow of 1/2 unit in each direction which is compatible with a total of one unit of flow entering and leaving each terminal. Another option is to require that for any orientation of the demand pairs, there is a feasible multicommodity for the pairs with one unit for each pair in the direction given by the orientation; however, deciding the routability according to this definition is not easy. For simplicity we require one unit of flow in each direction which results in a factor of 2 loss in the congestion when compared to other models.

an optimal upper bound of $\Theta(\log k)$ on multicommodity flow-cut gaps in undirected graphs in a variety of settings [20, 28, 18, 7]. Poly-logarithmic flow-cut gaps are also known in directed graphs with symmetric demand pairs [26]. In contrast to these results, the flow-cut gap in directed graphs can be polynomial [30, 15].

Maximum throughput routing problems, such as MEDP and the related problem of congestion minimization, aim to construct integer flows. These problems are typically tackled via relaxations based on multicommodity flows. Culminating a series of papers that addressed special cases and developed various tools, in a recent breakthrough, Chuzhoy [13] showed a poly-logarithmic upper bound with constant congestion in general undirected graphs. Subsequently, the congestion has been brought down to the optimal bound of 2 in [16]. Building on Chuzhoy's work, the authors of this paper obtained a poly-logarithmic upper bound with constant congestion for the maximum *node*-disjoint paths problem in undirected graphs. Our primary motivation is to understand whether the gap between fractional flows and integral flows is also small in directed graphs with symmetric demand pairs. We also believe that addressing this question will have auxiliary benefits that we discuss below.

Structure of graphs with large (directed) treewidth: Recent progress on routing problems has been accomplished via the following scheme. The well-linked decomposition framework of [9] showed that one can use flow-cut gap results to reduce the problem (to within poly-logarithmic factors) to a graph theoretic question: if the graph G has a "well-linked" set of size k, does it have a routing structure (called a *crossbar*) of size[3] $\tilde{\Omega}(k)$? For node-capacitated routing problems in undirected graphs, the question can be phrased in terms of the well-understood notion of *treewidth*: If G has treewidth k, does G have a crossbar of size $\tilde{\Omega}(k)$? The question was answered affirmatively (in [6], following Chuzhoy's framework for MEDP). The technical ingredients developed in [13], and in subsequent work [16, 6], have led to further graph theoretic results including a polynomial relationship between treewidth and the size of a largest grid-minor in a graph [5], and several applications [4].

An important motivation for studying routing problems in directed graphs with symmetric demand pairs stems from their connections to *directed treewidth*. Johnson, Robertson, Seymour, and Thomas [21] introduced the notion of directed treewidth. In undirected graphs treewidth is defined via tree decompositions and is reasonably intuitive. Directed treewidth is defined via arboreal decompositions [21] and is less easy to grasp. We refer the reader to [21, 29, 1] for some subtle issues that differentiate directed treewidth from treewidth. It is believed that understanding directed treewidth better would yield significant dividends in graph theory and algorithms. Fortunately, directed treewidth, like treewidth, can be approximately understood via well-linked sets [29]. In this paper, we extend the well-linked decomposition framework of [9] to this setting and this leads to the following question: If a directed graph G has directed treewidth k (equivalently, has a well-linked set of size

[3] The $\tilde{\Omega}$ notation hides poly-logarithmic factors.

$\Omega(k)$), does it have a "routing structure" of size $\tilde{\Omega}(k)$? Answering this question affirmatively would lead to algorithms for disjoint path routing for symmetric pair instances. In addition it may lead to insights about the relationship between treewidth and cylindrical minors in directed graphs; we refer the reader to [23, 24] for recent progress on this question.

Flow-cut gap in planar graphs: Another interesting direction for future work, and a motivation for us, is to show the following: if G is a planar directed graph with directed treewidth k, then it has a crossbar of size $\Omega(k)$. A linear relationship would have applications to routing on disjoint paths, and it would also give an improved upper bound on the flow-cut gap for symmetric product multicommodity flows in planar directed graphs. Currently, the flow-cut gap for product multicommodity flows in both general and planar directed graphs is known to be $O(\log n)$ [27]. The existence of a crossbar of size $\Omega(k)$ will imply that the flow-cut gap is $O(1)$ in planar directed graphs[4], which in turn will give constant factor approximation guarantees for problems such as the Uniform Sparsest Cut problem in planar directed graphs; such results are known for planar undirected graphs [25].

In this paper, we study the SymANF problem as a first step towards understanding maximum throughput routing problems in directed graphs. We now give a high-level description of our algorithm and we describe in more detail some specific technical contributions that enable us to prove Theorem 1.

2.1 Overview of the algorithm and technical contributions

Let $(G, \mathcal{M})$ be an instance of SymANF. Let $\mathcal{T}$ be the set of all nodes that participate in the pairs of $\mathcal{M}$; we refer to the nodes in $\mathcal{T}$ as the *terminals*. Our algorithm for SymANF in directed graphs follows the framework of Chekuri, Khanna, and Shepherd [8, 9] for the ANF problem in undirected graphs. In a nutshell, the framework decomposes an arbitrary instance of ANF into several instances that are *flow-well-linked*. The set of terminals $\mathcal{T} = \{s_1, t_1, \ldots, s_k, t_k\}$ is *flow-well-linked* if *any* matching on the terminals is routable. This is essentially equivalent (modulo a factor of 2 in congestion) to saying that G admits a symmetric product multicommodity flow where the weight on each terminal is 1 and is 0 on every non-terminal. If the terminals are flow-well-linked, we can route all the input pairs. Thus the heart of the matter is showing that an arbitrary instance can be decomposed into well-linked instances without losing too much flow.

The decomposition has two main components. The first step is a weaker decomposition in which we take a fractional solution to a natural multicommodity flow based LP (described in Section 3.3) and use it to decompose the instance into instances that are only *fractionally* flow-well-linked. More precisely, there is a weight function $\pi : \mathcal{T} \to [0, 1]$ and the terminals are

[4] The implications of crossbar results for product multicommodity flow-cut gaps is pointed out in [9].

flow-well-linked with respect to these weights; if all terminals have weight 1 then they are flow-well-linked. The second step is a clustering step in which we take a fractionally flow-well-linked instance and we identify a large subset of the pairs such that their endpoints are flow-well-linked. In this paper, we show how to implement these two steps for the SymANF problem in directed graphs. In the first step, we extend the approach of [9] to our setting; we refer the reader to Section 4 for the details of the decomposition. We note that the approximation factor that we lose in the decomposition is proportional to the flow-cut gap; for symmetric instances, the flow-cut gap is only $\text{polylog}(k)$. The second step poses several technical difficulties in directed graphs and it is our main technical contribution. We briefly highlight some of the difficulties involved in the clustering step, and we refer the reader to Section 5 for the details. Chekuri, Khanna, and Shepherd [9] gave a simple clustering technique for *edge-capacitated* undirected graphs. Roughly speaking, the approach is to take a spanning tree and to partition it into edge-disjoint subtrees where each subtree gathers roughly a unit weight from π. These subtrees are then used to find the desired flow-well-linked subset of pairs/terminals; one terminal is picked from each subtree. The clustering step is more involved in *node-capacitated* undirected graphs. The spanning tree approach, combined with some preprocessing to reduce the degree, gives a clustering for node-capacitated graphs with slightly weaker parameters [9]. In [10], the authors gave a stronger clustering for the node-capacitated setting; this approach is more involved than the spanning tree clustering and it exploits a connection between well-linked sets and treewidth; recent work [5] obtains a stronger result but requires more involved ideas. In directed graphs, there is no simple clustering process akin to using a spanning tree (or even an arborescence). Instead, our approach exploits the connection between well-linked sets and directed treewidth. However, the main challenge is to make this algorithmic. We also mention that, in addition to finding a large flow-well-linked set Y from a fractionally flow-well-linked set X, we also need to ensure that Y contains a large enough matching from the original set of pairs. For this purpose, we rely on a flow augmentation tool developed in [11]. These difficulties are also the reason why we are only able to obtain a constant congestion for SymANF while ANF admits a polylogarithmic ratio with congestion 1 in edge-capacitated graphs [8] and with congestion $(1 + \varepsilon)$ in node-capacitated graphs [9].

2.2 Discussion of related work

The ANF problem, the MEDP problem and its node-capacitated counterpart, the Maximum Node Disjoint Paths (MNDP) problem have been studied extensively in both undirected and directed graphs. We first discuss the decision versions of these problems where we are given G and the pairs, and the goal is to decide if all of them can be routed. It is easy to see that the decision version of ANF is polynomial-time solvable via linear programming — one needs to check whether there is a multicommodity flow that routes one unit of flow for each input pair. On the other hand, the decision versions of MEDP and MNDP, denoted by EDP and NDP respectively, are **NP**-complete if k is part of the input [22,17]. If k is fixed, Robertson and Seymour, building on their seminal work on graph minors, gave a polynomial-time algorithm for NDP (and hence also for EDP) in undirected graphs. Interestingly, EDP is already **NP**-complete for $k = 2$ in directed graphs [19]. It is useful to note that the undirected graph algorithm of Robertson and Seymour relies heavily on treewidth and the structure of graphs with large treewidth.

ANF, MEDP and MNDP are optimization problems. Although the decision version of ANF is poly-time solvable, ANF is **NP**-hard, and **APX**-hard to approximate, even in capacitated trees [20]; routing is trivial in trees, selecting the pairs to route is not. The best approximation guarantees that are known for the ANF problem in undirected graphs are an $O(\log^2 k)$ approximation in edge-capacitated graphs [9] and an $O(\log^4 k \log n)$ approximation with congestion $(1+\epsilon)$ in node-capacitated graphs [9]; these ratios improve by a logarithmic factor for planar graphs. For node-capacitated graphs, an unpublished manuscript [10] gives an $O(\log^2 k)$-approximation if constant congestion is allowed. The MEDP problem with congestion $c = o(\log n / \log \log n)$ is $\Omega\left((\log n)^{O(1/c)}\right)$-hard to approximate in undirected graphs [3] and $\Omega(n^{O(1/c)})$-hard to approximate in directed graphs [14], unless $\mathbf{NP} \subseteq \mathbf{ZPTIME}\left(n^{\mathrm{polylog}(n)}\right)$. It is useful to note that these hardness results also hold for the ANF problem, which suggests that the current techniques do not distinguish between the difficulty of ANF and MEDP. For MEDP in undirected graphs there is an $O(\sqrt{n})$-approximation with congestion 1 [11] and as we already mentioned, recent work obtains a polylogarithmic approximation with congestion 2 [16]. In directed graphs, MEDP has an $n^{O(1/c)}$-approximation with congestion c [31], and this approximation carries over to ANF as well. These approximation results use the natural multicommodity flow relaxations as a starting point and they also establish the same upper bound on the integrality gap of the relaxations.

Organization: Section 3 introduces the main definitions and technical tools that we use, and it describes the approximation algorithm for SymANF. Section 4 and Section 5 describe the well-linked decomposition and clustering technique for directed graphs with symmetric demand pairs.

3 Approximation Algorithm for SymANF

In the following, we work with an instance $(G, \mathcal{M})$ of the SymANF problem, where $G = (V, E)$ is a directed graph with unit node capacities and $\mathcal{M} = \{s_1t_1, \ldots, s_kt_k\}$ is a collection of node pairs. We refer to the nodes participating in the pairs of $\mathcal{M}$ as terminals, and we use $\mathcal{T}$ to denote the set of all terminals. We assume that the pairs $\mathcal{M}$ form a perfect matching on $\mathcal{T}$ and each terminal is a leaf in G, i.e., each terminal is connected to a single neighbor using an edge in each direction. One can reduce an arbitrary instance to an instance that satisfies these assumptions as follows. If a node v participates in several pairs, we make a copy of v for each of the pairs it participates in, and attach the copy v' to v using an edge in each direction; finally we replace v by v' in the pair. Similarly, if a terminal is not a leaf, we make a copy of the terminal, we attach the copy to the original node as a leaf, and we replace the terminal by its copy in the pairs that contain it. Note that, if a set of pairs was routable in the original instance, then it is routable in the new instance with congestion at most 2.

In the following subsection we describe some basic ingredients about (symmetric) multicommodity flows and node separators. Subsection 3.2 formally defines well-linked sets and states the results on well-linked decompositions and clustering that are used in the algorithm for SymANF.

3.1 Multicommodity flows and sparse node separators

Let $G = (V, E, \mathrm{cap})$ be a directed node-capacitated graph with node capacities given by cap. In this paper, we work with path-based flows f that assign a non-negative real value $f(p)$ to each path in G. A flow f is *feasible* if it satisfies the capacity constraints; more precisely, for each node v, $\sum_{p: v \in p} f(p) \leq \mathrm{cap}(v)$. For any ordered pair (u, v) of nodes, the total flow from u to v is $\sum_{p \in \mathcal{P}(u,v)} f(p)$, where $\mathcal{P}(u, v)$ is the set of all paths of G from u to v.

A *multicommodity flow* instance in G is a demand vector $\mathbf{d}$ that assigns a non-negative real value $d(u, v)$ to each ordered pair (u, v) of nodes of G; we refer to $d(u, v)$ as the demand of the pair (u, v). A multicommodity flow instance is *symmetric* if $d(u, v) = d(v, u)$ for all ordered pairs (u, v). A multicommodity flow instance $\mathbf{d}$ is a *product* multicommodity flow instance if $d(u, v) = w(u)w(v)$, where $w : V \to \mathbb{R}_+$ is a weight function on the nodes of G. Note that a product multicommodity flow instance is symmetric. In the following, we only consider symmetric multicommodity flow instances. A multicommodity flow instance $\mathbf{d}$ is *routable* if there is a feasible multicommodity flow in which, for each ordered pair (u, v), the total flow on the paths from u to v is at least $d(u, v)$. We work with the following quantities associated with a *symmetric* multicommodity flow instance, the maximum concurrent flow and the sparsest node separator. The *maximum concurrent flow* is the maximum value $\lambda \geq 0$ such that $\lambda\mathbf{d}$ is routable. A *node separator* is a set $C \subseteq V$ of nodes. The removal of a node separator gives us one or more strongly connected components; we say that a pair uv is separated by C if u and v are *not* in the same strongly connected component of $G - C$. The demand separated by C, denoted by $\mathrm{dem}_{\mathbf{d}}(C)$, is the total demand of all of the unordered pairs separated by C; more precisely, $\mathrm{dem}_{\mathbf{d}}(C) = \sum_{uv \text{ separated by } C} d(u, v)$. The *sparsity* of a node separator C is $\mathrm{cap}(C)/\mathrm{dem}_{\mathbf{d}}(C)$. A *sparsest node separator* is a separator with minimum sparsity. It is straightforward to verify that, for a symmetric multicommodity flow, the minimum sparsity of a node separator is an upper bound on the maximum concurrent flow. The *flow-cut gap* in G is the maximum value — over all symmetric multicommodity flow instances $\mathbf{d}$ in G — of the ratio between the minimum sparsity of a node separator and the maximum concurrent flow. The flow-cut gap in any graph is $O(\log^2 k)$, where k is the number of commodities (each pair (u,v) with non-zero demand is a commodity) [26]. For product multicommodity flows, the flow-cut gap is $O(\log k)$ [27]. Moreover, it was shown in [27] that there is a polynomial time algorithm that, given a product multicommodity flow instance $\mathbf{d}$ in G, it constructs a node separator C whose sparsity is at most $O(\log k)\lambda$, where λ is the maximum concurrent flow for $\mathbf{d}$; we use such an algorithm in a black box fashion in the well-linked decomposition step that we describe in more detail below.

A *node separation* in G is a partition (A,B,C) of the nodes of G such that there is no edge of G from A to B (note that there can be an edge of G from B to A). The following proposition shows that, given a weight function π on the nodes, and a node separator C, one can choose a node separation (A,B,C) that is "balanced" with respect to π.

Proposition 1 *Let $G=(V,E)$ be a directed graph and let $\pi : V \to \mathbb{R}_+$ be a weight function. Let $\mathbf{d}$ be the following product multicommodity flow: $d(u,v) = \pi(u)\pi(v)/\pi(V)$ for each pair (u,v) of nodes. Let C be a node separator in G. There is a node separation (A,B,C) such that $\mathrm{dem}_{\mathbf{d}}(C) \le 2\min\{\pi(A),\pi(B)\}$. Moreover, given C, we can compute such a node separation in polynomial time.*

Proof: Let C be a node separator. Let $K_1, K_2, \ldots, K_\ell$ be a topological ordering of the strongly connected components of $G-C$ in which each edge of $G-C$ connecting different strongly connected components is oriented from right to left.

Suppose that $\pi(K_i) \le \pi(V-C)/2$ for each i. Let p be the smallest index such that $\pi(K_1 \cup \cdots \cup K_p) \ge \pi(V-C)/4$. Let $A = V(K_1) \cup \cdots \cup V(K_p)$ and $B = V(K_{p+1}) \cup \cdots \cup V(K_\ell)$. Since $\pi(K_1 \cup \cdots \cup K_{p-1}) < \pi(V-C)/4$ and $\pi(K_p) \le \pi(V-C)/2$, we have $\pi(A) \le 3\pi(V-C)/4$ and therefore $\pi(B) \ge \pi(V-C)/4$. Thus (A,B,C) is a node separation satisfying $\min\{\pi(A),\pi(B)\} \ge \pi(V-C)/4$. Note that the total demand of the pairs $(u,v) \in (V-C)\times(V-C)$ is $\pi(V-C)\cdot\pi(V-C)/\pi(V) \le \pi(V-C)$. Therefore $\mathrm{dem}_{\mathbf{d}}(C) \le \pi(V-C)/2 \le 2\min\{\pi(A),\pi(B)\}$.

Therefore we may assume that $\max_i \pi(K_i) > \pi(V-C)/2$. Let K_q be the strongly connected component with maximum π-weight; more precisely, $q = \mathrm{argmax}_i \pi(K_i)$. We define a partition (A,B) of $V-C$ as follows. If $\pi(K_1 \cup \cdots \cup K_{q-1}) \ge \pi(K_{q+1} \cup \cdots \cup K_\ell)$, we let $A = V(K_1) \cup \cdots \cup V(K_{q-1})$ and $B = V(K_q) \cup \cdots \cup V(K_\ell)$. Otherwise, we let $A = V(K_1) \cup \cdots \cup V(K_q)$ and $B = V(K_{q+1}) \cup \cdots \cup V(K_\ell)$. The partition (A,B,C) is a node separation satisfying $\min\{\pi(A),\pi(B)\} \ge \pi(V-(C\cup K_q))/2$. Note that the total demand of the pairs $(u,v) \in (V-(C\cup K_q))\times(V-(C\cup K_q))$ is $\pi(V-(C\cup K_q))\cdot\pi(V-(C\cup K_q))/\pi(V)$. Additionally, the total demand of the pairs $(u,v) \in K_q \times (V-(C\cup K_q))$ is $\pi(K_q)\pi(V-(C\cup K_q))/\pi(V)$. Therefore we have

$$
\begin{aligned}
\mathrm{dem}_{\mathbf{d}}(C) &\le \frac{\pi(K_q)\pi(V-(C\cup K_q))}{\pi(V)} + \frac{\pi(V-(C\cup K_q))\pi(V-(C\cup K_q))}{2\pi(V)} \\
&= \frac{\pi(V-(C\cup K_q))(\pi(K_q)+\pi(V-C))}{2\pi(V)} \\
&\le \pi(V-(C\cup K_q))
\end{aligned}
$$

Therefore $\mathrm{dem}_{\mathbf{d}}(C) \leq 2 \min \{\pi(A), \pi(B)\}$. □

3.2 Well-linked sets, decomposition, and clustering

There are two notions of well-linkedness that have been used for routing problems in undirected graphs [9]; one is based on a flow requirement and the other is based on a cut requirement. In the following, we define directed node-capacitated versions of these two notions and we show some basic properties of these notions.

Flow-well-linked sets: Let G be a directed graph with unit capacities on the nodes. We define a fractional version of flow-well-linkedness as follows. Let $\pi : X \to [0, 1]$ be a weight function on $X \subseteq V$. Let $\mathbf{d}$ be the following demand vector: $d(u, v) = \pi(u)\pi(v)/\pi(X)$ for each ordered pair (u, v) of nodes in X. The set X is **π-flow-well-linked** in G iff $\mathbf{d}$ is routable in G. For a scalar $c \in [0, 1]$, we say that X is c-flow-well-linked if X is π-flow-well-linked, where $\pi(v) = c$ for each vertex $v \in X$.

Cut-well-linked sets: A set $X \subseteq V$ is **cut-well-linked** in G iff, for any two disjoint subsets Y and Z of X of equal size, there are $|Y|$ node-disjoint paths from Y to Z in G. Recall that a node is a leaf in G if it is connected to a single neighbor using an edge in each direction. If the nodes of X are leaves in G, an equivalent definition is the following. The set X is cut-well-linked iff, for any node separation (A, B, C) satisfying $X \cap C = \emptyset$, we have $|C| \geq \min \{|X \cap A|, |X \cap B|\}$. We define a fractional version of cut-well-linkedness as follows. Let X be a set of nodes of G and let $\pi : X \to [0, 1]$ be a weight function on X. Suppose that all the nodes in X are leaves of G. The set X is **π-cut-well-linked** in G if, for any node separation (A, B, C), we have $|C| \geq \min \{\pi(A), \pi(B)\}$. Note that, since the nodes in X are leaves, it suffices to check this condition for separations (A, B, C) for which $\pi(C) = 0$. Now consider a set X that contains nodes that are not leaves. For each node $x \in X$, we add a new node x' and connect x' to x using two edges, one in each direction. Let X' be the set of new nodes, let G' be the resulting graph, and let $\pi' : X' \to [0, 1]$ be the weight function $\pi'(x') = \pi(x)$ for each node $x \in X$. The set X is π-cut-well-linked in G iff X' is π'-cut-well-linked in G'.

The following proposition relates the two notions of well-linkedness.

Proposition 2 *Let $G = (V, E)$ be a directed graph. Let X be a set of nodes and let $\pi : X \to [0, 1]$ be a weight function on X. Let $\alpha = \alpha(G) \geq 1$ be an upper bound on the worst case flow-cut gap for product multicommodity flows in G. If X is π-flow-well-linked in G then X is $(\pi/2)$-cut-well-linked in G. If X is π-cut-well-linked in G then X is $(\pi/(2\alpha))$-flow-well-linked in G.*

Proof: Let $\mathbf{d}$ be the following product multicommodity flow: $d(u, v) = \pi(u)\pi(v)/\pi(X)$ for each pair (u, v) of nodes in X, and $d(\cdot)$ is zero for all other pairs.

Suppose that X is π-flow-well-linked. Recall that, in order to show that X is $(\pi/2)$-cut-well-linked, it suffices to verify that, for each node separation (A, B, C) such that $\pi(C) = 0$, we have $|C| \geq \min\{\pi(A), \pi(B)\}/2$. Consider a node separation (A, B, C) such that $\pi(C) = 0$. Since X is π-flow-well-linked, $\mathbf{d}$ is routable and therefore $|C| \geq \text{dem}_{\mathbf{d}}(C) = \pi(A)\pi(B)/\pi(X)$. Since $\pi(X) = \pi(A) + \pi(B)$, we have $\pi(A)\pi(B)/\pi(X) \geq \min\{\pi(A), \pi(B)\}/2$, as desired.

Conversely, suppose that X is π-cut-well-linked. By definition, X is $(\pi/(2\alpha))$-flow-well-linked if $\mathbf{d}/(2\alpha)$ is routable. Thus, in order to show that X is $(\pi/(2\alpha))$-flow-well-linked, it suffices to verify that each node separator has sparsity at least $1/2$.

Let C be a sparsest node separator. By Proposition 1, there is a node separation (A, B, C) such that $\text{dem}_{\mathbf{d}}(C) \leq 2\min\{\pi(A), \pi(B)\}$. Since X is π-cut-well-linked, we have $|C| \geq \min\{\pi(A), \pi(B)\}$. Therefore the sparsity of C is at least $1/2$. □

Well-linked decomposition: The following theorem is an extension to directed graphs of the well-linked decomposition technique introduced by [9] for routing problems in undirected graphs. The proof follows the outline of the approach in [9] and it can be found in Section 4.

Theorem 2 *Let* OPT *be the value of a solution to the* symANF-LP *relaxation*[5] *for a given instance $(G, \mathcal{M})$ of* SymANF. *Let $\alpha = \alpha(G) \geq 1$ be an upper bound on the worst case flow-cut gap for product multicommodity flows in G. There is a partition of G into node-disjoint induced subgraphs $G_1, G_2, \ldots, G_\ell$ and weight functions $\pi_i : V(G_i) \to \mathbb{R}_+$ with the following properties. Let $\mathcal{M}_i$ be the induced pairs of $\mathcal{M}$ in G_i and let X_i be the endpoints of the pairs in $\mathcal{M}_i$. We have*

(a) $\pi_i(u) = \pi_i(v)$ *for each pair* $uv \in \mathcal{M}_i$.
(b) X_i *is* π_i*-flow-well-linked in* G_i.
(c) $\sum_{i=1}^{\ell} \pi_i(X_i) = \Omega(\text{OPT}/(\alpha \log \text{OPT})) = \Omega(\text{OPT}/\log^2 k)$.

Moreover, such a partition is computable in polynomial time if there is a polynomial time algorithm for computing a node separator with sparsity at most $\alpha(G)$ times the maximum concurrent flow.

From fractional well-linked sets to well-linked sets: The following theorem describes an algorithm that obtains a well-linked set from a fractionally well-linked set. The proof is given in Section 5.

Theorem 3 *Let X be a π-flow-well-linked set in G and let $\mathcal{M}$ be a perfect matching on X such that $\pi(u) = \pi(v)$ for each pair $uv \in \mathcal{M}$. There is a matching $\mathcal{M}' \subseteq \mathcal{M}$ on a set $X' \subseteq X$ such that X' is $1/32$-flow-well-linked in G and $|\mathcal{M}'| = 2|X'| = \Omega(\pi(X))$. Moreover, given X and $\mathcal{M}$, we can construct X' and $\mathcal{M}'$ in polynomial time.*

[5] The symANF-LP relaxation is given in Subsection 3.3.

Routing a flow-well-linked instance: Finally, we observe that, if an instance of SymANF is c-flow-well-linked for some $c \leq 1$, then we can route all of the pairs with congestion at most $2/c$.

Proposition 3 *Let $(G, \mathcal{M})$ be an instance of* SymANF *and let X be the set of all vertices that participate in the pairs of $\mathcal{M}$. If X is c-flow-well-linked for some $c \leq 1$, then we can route all of the pairs of $\mathcal{M}$ with congestion at most $2/c$.*

Proof: Note that it suffices to show that we can route c units of flow for each pair using congestion at most 2; once we have this flow, we can simply scale it by $1/c$ to get a flow that routes one unit of flow for each pair.

Let X_1 be a set consisting of exactly one node from each pair of $\mathcal{M}$, and let $X_2 = X - X_1$ be the set of all partners of the nodes in X_1. Let $\mathbf{d}$ be the following demand vector: $d(u, v) = c/|X|$ for each pair (u, v) of nodes in X, and $d(\cdot)$ is zero for all other pairs. Since X is c-flow-well-linked, there is a feasible flow f that routes $\mathbf{d}$. Note that f gives us a feasible flow in which each node in X_1 sends c units of flow to its partner: consider a node $u \in X_1$ and let v be its partner; we combine the flow paths of f connecting u to X and the flow paths of f connecting X to v in order to get flow paths from u to v carrying at least c units of flow. Similarly, f also gives us a feasible flow in which each node in X_2 sends c units of flow to its partner. The sum of the two flows gives us a congestion two flow that routes c units of flow for each pair of $\mathcal{M}$. □

3.3 The approximation algorithm for SymANF

We now describe our algorithm for SymANF. Let $(G, \mathcal{M})$ be an instance of SymANF. We consider a natural multicommodity flow relaxation for the problem. For each ordered pair (u, v) of nodes of G, let $\mathcal{P}(u, v)$ be the set of all paths in G from u to v. Since $\mathcal{M}$ forms a matching on $\mathcal{T}$, for all $i \neq j$, the sets $\mathcal{P}(s_i, t_i)$, $\mathcal{P}(t_i, s_i)$, $\mathcal{P}(s_j, t_j)$, and $\mathcal{P}(t_j, s_j)$ are pairwise disjoint. Let $\mathcal{P} = \bigcup_{i=1}^{k} (\mathcal{P}(s_i, t_i) \cup \mathcal{P}(t_i, s_i))$. For each path $p \in \mathcal{P}$, we have a variable $f(p)$ that is equal to the amount of flow on p. For each unordered pair $s_i t_i \in \mathcal{M}$ we have a variable x_i to indicate whether to route the pair or not. The LP relaxation ensures the symmetry constraint: there is a flow from s_i to t_i of value x_i *and* a flow from t_i to s_i of value x_i. Recall that we will be working with the node-capacitated problem and each node has unit capacity.

(symANF-LP)

$$
\begin{array}{lll}
\max & \displaystyle\sum_{i=1}^{k} x_i & \\
\text{s.t.} & \displaystyle\sum_{p \in \mathcal{P}(s_i,t_i)} f(p) \geq x_i & 1 \leq i \leq k \\
& \displaystyle\sum_{p \in \mathcal{P}(t_i,s_i)} f(p) \geq x_i & 1 \leq i \leq k \\
& \displaystyle\sum_{p\colon v \in p} f(p) \leq 1 & v \in V(G) \\
& x_i \leq 1 & 1 \leq i \leq k \\
& f(p) \geq 0 & p \in \mathcal{P}
\end{array}
$$

The dual of the symANF-LP relaxation has polynomially many variables and exponentially many constraints. The separation oracle for the dual is the shortest path problem. Thus we can solve the relaxation in polynomial time. Alternatively, we can write an equivalent LP relaxation that is polynomial sized.

The algorithm is described below.

(1) Solve the relaxation symANF-LP to get an optimal fractional solution (x, f) for the instance $(G, \mathcal{M})$.
(2) Use the well-linked decomposition (Theorem 2) to get a collection $(G_1, \mathcal{M}_1, \pi_1)$, ..., $(G_\ell, \mathcal{M}_\ell, \pi_\ell)$ of disjoint instances and weight functions.
(3) For each instance $(G_i, \mathcal{M}_i, \pi_i)$ in the decomposition, use the clustering technique (Theorem 3) to get an instance $(G_i, \mathcal{M}'_i)$.
(4) For each instance $(G_i, \mathcal{M}'_i)$, route all of the pairs of $\mathcal{M}'_i$ in G_i (Proposition 3). Output the union of these routings.

Let OPT be the value of the symANF-LP for the given instance, which lower bounds the number of pairs routed in an optimum solution. Combining Theorems 2 and 3 and Proposition 3, the number of pairs routed by the algorithm is $\sum_{i=1}^{\ell} |\mathcal{M}'_i| = \sum_{i=1}^{\ell} \Omega(\pi(V(\mathcal{M}_i))) = \Omega(\mathrm{OPT}/\log^2 k)$. Since each instance $(G_i, \mathcal{M}'_i)$ is 1/32-flow-well-linked, the routing in G_i has congestion at most 64. The graphs $G_1, \ldots, G_\ell$ are node disjoint and hence the pairs routed in these graphs do not interfere with each other. This completes the proof of Theorem 1.

4 Well-linked decomposition

In this section, we prove Theorem 2. We follow the notation and the approach introduced in [9] for edge and node-capacitated multicommodity flow problems in undirected graphs.

Let (x, f) be a solution to the symANF-LP with value $\mathrm{OPT} = \sum_{i=1}^{k} x_i$. The flow f is a symmetric multicommodity flow; as before, we view f as a path-based flow. Let H be a node-induced subgraph of G. For each ordered pair

Decomposition Algorithm

Input: Strongly connected subgraph H.
Output: Node-disjoint subgraphs $H_1, H_2, \ldots, H_\ell$ with associated weight functions $\pi_1, \pi_2, \ldots, \pi_\ell$, where each H_i is a node-induced subgraph of H.

(1) Suppose that $0 < \mathbf{w}(H) \leq \alpha \log \mathrm{OPT}$. Let $\pi(u) = w(u; H)/(8\alpha \log \mathrm{OPT})$ for each node $u \in V(H)$. Stop and output H and π.
(2) Suppose that $\mathbf{w}(H) > \alpha \log \mathrm{OPT}$. Let $\mathbf{d}$ be the following demand vector: $d(u,v) = w(u;H)w(v;H)/\mathbf{w}(H)$ for each ordered pair (u,v) of nodes in H. Let λ be the maximum concurrent flow for $\mathbf{d}$.
  (a) If $\lambda \geq 1/(8\alpha \log \mathrm{OPT})$, stop the recursive procedure. Let $\pi(u) = w(u;H)/(8\alpha \log \mathrm{OPT})$ for each node $u \in V(H)$. Output H and π.
  (b) Otherwise find a node separation (A, B, C) such that $|C| \leq \min\left\{\sum_{a \in A} w(a; H), \sum_{b \in B} w(b; H)\right\}/(4 \log \mathrm{OPT})$. Recursively decompose each strongly connected component of $H - C$. Output the decompositions of the strongly connected components.

(u,v) of nodes in H, let $\gamma(u,v;H)$ be the total amount of f-flow on paths p from u to v that are completely contained in H. For each unordered pair uv of nodes in H, let $\gamma'(u,v;H) = \gamma'(v,u;H) = \min\{\gamma(u,v;H), \gamma(v,u;H)\}$. For each node u in H, let $w(u;H) = \sum_{v \in V(H)} \gamma'(u,v;H)$. Let $\mathbf{w}(H) = \sum_{u \in V(H)} w(u;H)$.

We will need the following observation. Recall that a node separation in G is a partition (A,B,C) of the nodes of G such that there is no edge of G from A to B.

Proposition 4 *Let $G = (V,E)$ be a directed graph. Let $\pi : V \to [0,1]$ be a weight function. Let $\alpha = \alpha(G) \geq 1$ be an upper bound on the worst case flow-cut gap for product multicommodity flows in G. Suppose that V is not π-flow-well-linked in G. There is a node separation (A,B,C) such that $|C| \leq 2\alpha \min\{\pi(A), \pi(B)\}$. Moreover, we can construct such a separation in polynomial time if there is a polynomial time algorithm for computing a node separator with sparsity at most α times the maximum concurrent flow.*

Proof: Note that it follows from Proposition 2 that, for any weight function $\pi : V \to [0,1]$, either V is π-flow-well-linked or there is a node separation (A,B,C) such that $|C| \leq 2\alpha \min\{\pi(A), \pi(B)\}$. Additionally, we can construct such a separation in polynomial time as follows.

Let $\mathbf{d}$ be the following demand vector: $d(u,v) = \pi(u)\pi(v)/\pi(V)$ for each ordered pair (u,v) of nodes. Since $\mathbf{d}$ is not routable, we can compute in polynomial time a node separator C such that $|C| \leq \alpha\ \mathrm{dem}_{\mathbf{d}}(C)$. By Proposition 1, once we have C, we can compute in polynomial time a node separation (A,B,C) such that $\mathrm{dem}_{\mathbf{d}}(C) \leq 2\min\{\pi(A), \pi(B)\}$. The resulting separation (A,B,C) satisfies $|C| \leq 2\alpha \min\{\pi(A), \pi(B)\}$. □

Note that, in the step ($2b$) of the algorithm, we used Proposition 4: let $\pi(u) = w(u;H)/(8\alpha \log \mathrm{OPT})$ for each node u in H; since $\lambda < 1/(8\alpha \log \mathrm{OPT})$, $V(H)$

is not π-flow-well-linked and therefore there is a node separation (A, B, C) such that

$$|C| \leq 2\alpha \min\{\pi(A), \pi(B)\} = \min\left\{\sum_{a\in A} w(a; H), \sum_{b\in B} w(b; H)\right\} / (4 \log \mathrm{OPT}).$$

We apply the decomposition algorithm to each strongly connected component of G in order to get a decomposition of G into node-induced disjoint subgraphs $G_1, G_2, \dots, G_\ell$ with associated weight functions $\pi_1, \pi_2, \dots, \pi_\ell$. In the following, we show that this decomposition has the properties required by Theorem 2. It is straightforward to verify that the decomposition has the first two properties and thus we focus on the third property.

From the terminating conditions, it follows that $\pi_i(G_i) \geq \mathbf{w}(G_i)/(8\alpha \log \mathrm{OPT})$ for each i. Therefore it suffices to show that $\sum_{i=1}^k \mathbf{w}(G_i) \geq \mathbf{w}(G)/2 = \mathrm{OPT}/2$. Equivalently, the total flow lost is at most $\mathbf{w}(G)/2$, where the flow lost is $\mathbf{w}(G) - \sum_{i=1}^k \mathbf{w}(G_i)$.

We upper bound the total flow lost as follows. We say that a node of G was cut in the decomposition if the node belongs to a node separator C found in the step $(2b)$. We first note that the total flow lost is at most twice the number of nodes that were cut by the decomposition. We can show this as follows. Let Z be the set of all nodes that are cut by the decomposition. Recall that, for each pair uv of nodes, the amount of f-flow from u to v is equal to the amount of f-flow from v to u; we think of the flow from u to v and the flow from v to u as partner flows. Now consider the f-flow that does not contribute to $\sum_{i=1}^k \mathbf{w}(G_i)$: this flow can be partitioned into flows, each of which is on paths that intersect Z or it is the partner of a flow whose paths intersect Z. Since each node has unit capacity, the total f-flow on paths that intersect Z is at most $|Z|$ and thus the total flow lost is at most $2|Z|$. Thus it suffices to show that $|Z|$ is at most $\mathbf{w}(G)/4$.

Lemma 1 *The number of nodes cut by the well-linked decomposition is at most* $\mathbf{w}(G)/4$.

Proof: We charge the cut nodes as follows. Consider an iteration of the decomposition algorithm that cuts a node. Let H denote the graph considered in the current iteration and let (A, B, C) be the node separation found in Step $(2b)$. Recall that we have

$$|C| \leq \frac{1}{4 \log \mathrm{OPT}} \min\left\{\sum_{a\in A} w(a; H), \sum_{b\in B} w(b; H)\right\}.$$

We charge the nodes in C as follows. Let $D = A$ if $\sum_{a\in A} w(a; H) \leq \sum_{b\in B} w(b; H)$ and $D = B$ otherwise. We refer to D as the smaller side of the separation (A, B, C). For each node $u \in D$, we charge $w(u; H)/(4 \log \mathrm{OPT})$ to u. Note that the total charge to the nodes of D is $\sum_{u\in D} w(u; H)/(4 \log \mathrm{OPT}) \geq |C|$.

The total amount charged by the charging scheme is at least the number of nodes that are cut by the decomposition and thus it suffices to upper bound the total amount charged. We can show that the total amount charged is at most $\mathbf{w}(G)/4$ as follows. For each node u, we claim that u is charged at most $w(u; G)/4$. A node u is charged only if it is on the smaller side of the separation found in Step ($2b$) and therefore it is charged at most $\log(\mathbf{w}(G)) = \log \mathrm{OPT}$ times. Additionally, each charge to u is at most $w(u; G)/(4 \log \mathrm{OPT})$. □

5 From fractional well-linked sets to well-linked sets

In this section, we prove Theorem 3. We prove the theorem in two steps. In the first step, we show that there exists a set Y of cardinality $\Omega(\pi(X))$ such that Y is $\Omega(1)$-flow-well-linked. Additionally, the set Y can send flow to X and receive flow from X. In the second step, we use Y to select a matching $\mathcal{M}' \subseteq \mathcal{M}$ of size $\Omega(|Y|)$.

Before we give the details of this procedure, we first give an intuitive (and non-constructive) argument that motivates the approach. The argument is partly inspired by the work in [10] and it differs from the low-degree spanning tree clustering that has been the main approach in the undirected case. The reader can skip the following paragraph and go straight to the technical proof.

Intuitive argument: In the following, we give an informal argument that illustrates the clustering for fractionally cut-well-linked sets. Suppose that G has a set X that is π-cut-well-linked. Recall that the directed treewidth of G is within a constant factor of the largest cut-well-linked set in G; this approximate duality relation is shown via the notion of *havens* [21]. Using a similar argument, one can show that, if X is π-cut-well-linked in G, the directed treewidth of G is $\Omega(\pi(X))$. By applying the approximate duality again, we get that there is a cut-well-linked Z in G of size $|Z| = \Omega(\pi(X))$. Since the existence of Z was shown via the π-cut-well-linkedness of X, it is intuitive that there is such a set Z that is reachable from X in the following sense: there is a (single commodity) flow from X to Z where each node in Z receives one unit of flow and each node v in X sends $\pi(v)$ units of flow; similarly there is a flow from Z to X. The existence of these flows together with the fact that X is π-cut-well-linked imply that Z is $\Omega(1)$-cut-well-linked. We then have to identify a subset $X' \subset X$ that is $\Omega(1)$-cut-well-linked. Moreover, for the SymANF problem, we need to ensure that for the initial matching $\mathcal{M}$ on X there is a sufficiently large sub-matching of $\mathcal{M}$ induced on X'. These latter arguments require an incremental flow-augmentation technique from [11]. The main technical challenge is to *efficiently* find a Z reachable from X as described above. Surprisingly, we are able to show that a simple greedy iterative approach based on the intuition of the existence argument, with a careful argument, works to give the desired set Z modulo constant congestion. We believe that this is a useful technical building block for further work in this area. Now we give the formal argument.

First step: Finding a large well-linked set. In the first step, we find a set Y with the following properties:

Theorem 4 *Let G be a directed graph with unit node capacities. Let X be a set of nodes of G and let $\pi : X \to (0,1]$ be a weight function on X. Suppose that X is π-flow-well-linked in G. There is a polynomial time algorithm that constructs a set $Y \subseteq V(G)$ with the following properties.*

(P_1) $|Y| = \lfloor \pi(X)/8 \rfloor$.
(P_2) *Y is 1/4-flow-well-linked in G.*

Additionally, for any subset $X' \subseteq X$ such that $\pi(X') \leq \pi(X)/15$, we have

(Q_1) *There is a single commodity flow in G from X' to Y such that each node $x \in X'$ sends $\pi(x)/64$ units of flow and each node in Y receives at most one unit of flow.*
(Q_2) *There is a single commodity flow in G from Y to X' such that each node $x \in X'$ receives $\pi(x)/64$ units of flow and each node in Y sends at most one unit of flow.*

The main ingredient in the proof of Theorem 4 is the following lemma. The lemma shows that, if we have a set X that is π-flow-well-linked, then there exists a set Y of size $\Omega(\pi(X))$ such that Y is $\Omega(1)$-flow-well-linked. The main idea behind the lemma is the following. If X is π-cut-well-linked and Z is a node separator of size less than $\pi(X)/4$, there is a unique strongly connected component $\beta(Z)$ of $G - Z$ whose π-weight is more than half the weight of X. The main insight is that, if we consider the set Y of size $\lfloor \pi(X)/4 \rfloor$ for which $|Y \cup \beta(Y)|$ is minimum, this gives us the desired set. This gives us a non-constructive proof of the existence of such a set Y. Using a simple iterative procedure, we can find such a set Y in polynomial time.

Lemma 2 *Let G be a directed graph with unit node capacities. Let X be a set of nodes of G and let $\pi : X \to (0,1]$ be a weight function on X. Suppose that X is π-cut-well-linked in G. There is a polynomial time algorithm that constructs a set $Y \subseteq V(G)$ with the following properties.*

(R_1) $|Y| = \lfloor \pi(X)/4 \rfloor$.
(R_2) *There is a single commodity flow in G from X to Y such that each node $x \in X$ sends at most $\pi(x)$ units of flow and each node in Y receives one unit of flow.*
(R_3) *There is a single commodity flow in G from Y to X such that each node in Y sends one unit of flow and each node $x \in X$ receives at most $\pi(x)$ units of flow.*

We will need the following simple observation.

Proposition 5 *Let G be a directed graph. Let X be a set of nodes of G and let $\pi : X \to [0,1]$ be a weight function on X. Suppose that X is π-cut-well-linked in G. Then for any set Z such that $|Z| < \pi(X)/4$, there is a unique strongly connected component $\beta(Z)$ of $G - Z$ such that $\pi(\beta(Z)) > \pi(X)/2$.*

Proof: Suppose for contradiction that there is a set Z such that $|Z| < \pi(X)/4$ and, for each strongly component H of $G - Z$, we have $\pi(H) \leq \pi(X)/2$. Let $H_1, H_2, \ldots, H_\ell$ be a topological ordering of the strongly connected components of $G-Z$ in which each edge of $G-Z$ that connects different strongly connected components is oriented from right to left. Let p be the smallest index such that $\pi(H_1 \cup \cdots \cup H_p) \geq \pi(X)/4$. Note that, since $\pi(H_1 \cup \cdots \cup H_{p-1}) < \pi(X)/4$ and $\pi(H_p) \leq \pi(X)/2$, we have $\pi(H_1 \cup \cdots \cup H_p) < 3\pi(X)/4$. Thus we have $\pi(H_{p+1} \cup \cdots \cup H_\ell) > \pi(X)/4$. Let A be the set of all vertices in $H_1 \cup \cdots \cup H_p$ and let B be the set of all vertices in $H_{p+1} \cup \cdots \cup H_\ell$. Note that (A, B, Z) is a node separation in G. Since X is π-cut-well-linked, it follows that $|Z| \geq \min\{\pi(A), \pi(B)\} \geq \pi(X)/4$, which is a contradiction. □

Proof of Lemma 2: We start by introducing some notation. If X is a π-cut-well-linked set in G, it follows from Proposition 5 that, for each set $Z \subseteq V(G)$ such that $|Z| < \pi(X)/4$, there is a unique strongly connected component $\beta(Z)$ of $G - Z$ such that $\pi(\beta(Z)) > \pi(X)/2$.

We will maintain a set Y satisfying the first condition. If Y does not satisfy the second or the third condition, we show that we can find a set Y' satisfying the first condition such that $|Y' \cup \beta(Y')| < |Y \cup \beta(Y)|$. Initially, Y is an arbitrary subset of size $\lfloor \pi(X)/4 \rfloor$.

Suppose that Y does not satisfy the second condition. Let H_1 be the following network. We start with $H_1 = G$; recall that each node in G has a capacity of one. For each node $x \in X$, we add a node x' to H_1 and an edge from x' to x; the node x' receives a capacity of $\pi(x)$. We add a source node s and a directed edge from s to each node x'. We add a sink node t and an edge from each node in Y to t.

Consider the network H_1 and let X' be the set of all copies of the nodes in X. A triple (A, B, C) is an s-t separation in H_1 if the sets A, B, C partition $V(H_1)$, $s \in A$, $t \in B$, and there is no edge of H_1 from A to B. The capacity of a separation (A, B, C) is the capacity of the nodes in C. Let (A, B, C) be an s-t separation in H_1 with minimum capacity. Since Y does not satisfy the second condition, the capacity of C is smaller than $|Y|$. Let $A' = A - (X' \cup \{s\})$, $B' = B - (X' \cup \{t\})$, and $C' = C - X'$. Since t is in B and there is no edge of H_1 from A to B, we have $Y \subseteq B' \cup C'$.

In the following, we show that $\beta(C') \subseteq A' \cap \beta(Y)$. Since there is no edge of H_1 from A to B, for each node $x \in X \cap B$, we have $x' \in C$: if x' is in A, the edge from x' to x is connecting A to B; if x' is in B, the edge from s to x' is connecting A to B. Therefore $\text{cap}(C) \geq \pi(B) = \pi(B')$ and thus $\pi(B') \leq \pi(X)/4$. Since $\beta(C')$ is a strongly connected component of $G - C'$ and there is no edge of G from A' to B', we have that $\beta(C')$ is completely contained in one of A' and B'. Since $\pi(\beta(C')) > \pi(X)/2$ and $\pi(B') \leq \pi(X)/2$, we have $\beta(C') \subseteq A'$. Since $\beta(C')$ is contained in A', $\beta(C')$ is a strongly connected subgraph of $G - (B' \cup C')$. Since $Y \subseteq B' \cup C'$, there is a unique strongly connected component K of $G - Y$ that contains $\beta(C')$. Since $\beta(C')$ and $\beta(Y)$ overlap at a vertex of X, we have $K = \beta(Y)$. Therefore $\beta(C') \subseteq \beta(Y)$ and thus $\beta(C') \subseteq A' \cap \beta(Y)$, as claimed.

Since $|C'| < |Y|$, we have $|C' \cup \beta(C')| = |C'| + |\beta(C')| < |Y| + |\beta(Y)| = |Y \cup \beta(Y)|$. We let Y' be the set consisting of C' together with an arbitrary subset of $\beta(C')$ of size $\lfloor \pi(X)/4 \rfloor - |C'|$. Then Y' is the desired set.

Therefore we may assume that Y does not satisfy the third condition. The argument is very similar to the previous case, and we include it for completeness. Let H_2 be the following network. We start with $H_2 = G$. We add a source node s and a directed edge from s to each node in Y. For each node $x \in X$, we add a node x' to H_2 and an edge from x to x'; the node x' receives a capacity of $\pi(x)$. We add a sink node t and a directed edge from each node x' to t.

Consider the network H_2 and let X' be the set of all copies of the nodes in X. A triple (A, B, C) is an s-t separation in H_2 if the sets A, B, C partition $V(H_2)$, $s \in A$, $t \in B$, and there is no edge of H_2 from A to B. The capacity of a separation (A, B, C) is the capacity of the nodes in C. Let (A, B, C) be an s-t separation in H_2 with minimum capacity. Since Y does not satisfy the third condition, the capacity of C is smaller than $|Y|$. Let $A' = A - (X' \cup \{s\})$, $B' = B - (X' \cup \{t\})$, and $C' = C - X'$. Since s is in A there is no edge of H_2 from A to B, we have $Y \subseteq A' \cup C'$.

In the following, we show that $\beta(C') \subseteq B' \cap \beta(Y)$. Since there is no edge of H_2 from A to B, for each node $x \in X \cap A$, we have $x' \in C$: if x' is in A, the edge from x' to t is connecting A to B; if x' is in B, the edge from x to x' is connecting A to B. Therefore $\text{cap}(C) \geq \pi(A) = \pi(A')$ and thus $\pi(A') \leq \pi(X)/4$. Since $\beta(C')$ is a strongly connected component of $G - C'$ and there is no edge of G from A' to B', we have that $\beta(C')$ is completely contained in one of A' and B'. Since $\pi(\beta(C')) > \pi(X)/2$ and $\pi(A') \leq \pi(X)/2$, we have $\beta(C') \subseteq B'$. Since $\beta(C')$ is contained in B', $\beta(C')$ is a strongly connected subgraph of $G - (A' \cup C')$. Since $Y \subseteq A' \cup C'$, there is a unique strongly connected component K of $G - Y$ that contains $\beta(C')$. Since $\beta(C')$ and $\beta(Y)$ overlap at a vertex in X, we have $K = \beta(Y)$. Therefore $\beta(C') \subseteq \beta(Y)$ and thus $\beta(C') \subseteq B' \cap \beta(Y)$. Since $|C'| < |Y|$, we have $|C' \cup \beta(C')| = |C'| + |\beta(C')| < |Y| + |\beta(Y)| = |Y \cup \beta(Y)|$. We let Y' be the set consisting of C' together with an arbitrary subset of $\beta(C')$ of size $\lfloor \pi(X)/4 \rfloor - |C'|$. Then Y' is the desired set. □

Lemma 3 *Let $G = (V, E)$ be a node-capacitated directed network. Let A and B be two sets of nodes in G. Let $\pi : A \to \mathbb{R}_+$ and $\pi' : B \to \mathbb{R}_+$ be two weight functions. Suppose that A and B satisfy the following conditions:*

- *A is π-flow-well-linked.*
- *There is a feasible single-commodity flow f_1 in G from B to A such that each node $b \in B$ sends $\pi'(b)$ units of flow to A and each node $a \in A$ receives at most $\pi(a)$ units of flow.*
- *There is a feasible single-commodity flow f_2 in G from A to B such that each node $a \in A$ sends at most $\pi(a)$ units of flow and each node $b \in B$ receives $\pi'(b)$ units of flow.*

Then B is $(\pi'/4)$-flow-well-linked in G.

Proof: Let $\mathbf{d}_1$ be the following multicommodity flow instance: $d_1(b, a) = \pi(a)\pi'(b)/\pi(A)$ for each pair $(b, a) \in B \times A$, and $\mathbf{d}_1(\cdot)$ is zero for all other pairs. We claim that we can route $\mathbf{d}_1$ using congestion at most two. In order to prove the claim, we combine the flow f_1 and the flow f that routes the following product multicommodity flow instance $\mathbf{d}$: $d(a,a') = \pi(a)\pi(a')/\pi(A)$ for all pairs of nodes $(a,a') \in A \times A$. Let $F_1(b,a)$ be the amount of flow sent by f_1 from b to a. We split the flow of f_1 from b to a among the nodes of A as follows: for each node $a' \in A$, the amount of f_1-flow from b to a that we allocate to a' is $F_1(b,a)\pi(a')/\pi(A)$. We split the flow of f from a to a' among the nodes of B as follows: for each node $b \in B$, the amount of f-flow from a to a' that we allocate to b is $F_1(b,a)\pi(a')/\pi(A)$; since $\sum_b F_1(b,a) = \pi(a)\pi'(B)/\pi(A) \leq \pi(a)$, there is enough f-flow from a to a' to allocate to B. Finally, we concatenate the allocated flow paths as follows. Consider a node $b \in B$ and two nodes $a, a' \in A$. We allocated $F_1(b,a)\pi(a')/\pi(A)$ units of f_1-flow to a'; we can represent the allocated flow as a collection $\{(P_i, \epsilon_i)\}$, where P_i is a path from b to a and ϵ_i is the amount of f_1-flow on P_i that we allocated. We allocated $F_1(b,a)\pi(a')/\pi(A)$ units of f-flow to a'; we can represent the allocated flow as a collection $\{(Q_j, \delta_j)\}$, where Q_j is a path from a to a' and δ_j is the amount of f-flow on Q_j that we allocated. By making multiple copies of each path, we may assume that $\epsilon_i = \delta_j = \epsilon$ for all i and j; that is, all flow paths have the same amount ϵ of flow. For each i, we send ϵ units of flow on the path obtained by concatenating P_i and Q_i; more precisely, we replace the flow paths $\{(P_i, \epsilon)\}$ and $\{(Q_i, \epsilon)\}$ by the flow paths $\{(P_i Q_i, \epsilon)\}$. By concatenating all of the allocated flow paths, we get a flow with congestion at most two. For each pair $(b, a') \in B \times A$, the amount of flow from b to a' is $\sum_{a \in A} F_1(b,a)\pi(a')/\pi(A) = \pi'(b)\pi(a')/\pi(A) = d_1(b,a')$.

Let $\mathbf{d}_2$ be the following multicommodity flow instance: $d_2(a,b) = \pi(a)\pi'(b)/\pi(A)$ for each pair $(a,b) \in A \times B$, and $d_2(\cdot)$ is zero for all other pairs. By combining the flows f_2 and f, we can show that $\mathbf{d}_2$ is routable with congestion at most two; the argument is very similar to the previous argument and we omit it.

Let g_1 and g_2 be the congestion two flows that route $\mathbf{d}_1$ and $\mathbf{d}_2$, respectively. In the following, we show how to combine g_1 and g_2 to get a congestion four flow that routes the following product multicommodity flow instance $\mathbf{d}'$: $d'(b,b') = \pi'(b)\pi'(b')/\pi'(B)$ for each pair of nodes $(b,b') \in B \times B$. Consider a node $a \in A$ and two nodes $b_1, b_2 \in B$. The amount of g_1-flow from b_1 to a is $\pi(a)\pi'(b_1)/\pi(A)$; we allocate $\pi(a)\pi'(b_1)\pi'(b_2)/(\pi(A)\pi'(B))$ of this flow to b_2. The amount of g_2-flow from a to b_2 is $\pi(a)\pi'(b_2)/\pi(A)$; we allocate $\pi(a)\pi'(b_1)\pi'(b_2)/(\pi(A)\pi'(B))$ of this flow to b_1. By concatenating the allocated flow paths, we can send $\pi(a)\pi'(b_1)\pi'(b_2)(\pi(A)\pi'(B))$ units of flow from b_1 to b_2 through a; summing over all nodes $a \in A$, the total flow from b_1 to b_2 is $\pi'(b_1)\pi'(b_2)/\pi'(B)$. Therefore $\mathbf{d}'$ is routable with congestion at most four. Thus B is $(\pi'/4)$-flow-well-linked. $\square$

Now we are ready to prove Theorem 4.

Proof of Theorem 4: Since X is π-flow-well-linked in G, it follows from Proposition 2 that X is $(\pi/2)$-cut-well-linked in G. By Lemma 2, there is a set Y with the following properties.

- $|Y| = \lfloor \pi(X)/8 \rfloor$.
- There is a single commodity flow f_1 in G from X to Y such that each node $x \in X$ sends at most $\pi(x)/2$ units of flow and each node in Y receives one unit of flow.
- There is a single commodity flow f_2 in G from Y to X such that each node in Y sends one unit of flow and each node $x \in X$ receives at most $\pi(x)/2$ units of flow.

By Lemma 3, Y is 1/4-flow-well-linked; here we applied the lemma with $A = X$, $B = Y$, $\pi(x) = \pi(x)$ for each $x \in X$, and $\pi'(y) = 1$ for each $y \in Y$.

Let $X_1 \subseteq X$ be the set of all nodes $x \in X$ such that x sends at least $\pi(x)/32$ units of flow in f_1. We can show that $\pi(X_1) \geq \pi(X)/15$ as follows. For a set $A \subseteq X$, let $F_1(A)$ be the total amount of f_1-flow sent by the nodes in A. We have $F_1(X) = |Y| \geq \pi(X)/16$. Additionally, since each node $x \in X - X_1$ sends at most $\pi(x)/32$ units of flow in f_1, we have $F_1(X - X_1) \leq (\pi(X) - \pi(X_1))/32$. It follows that $F_1(X_1) \geq (\pi(X) + \pi(X_1))/32$ and therefore $\pi(X_1)/2 \geq F_1(X_1) \geq (\pi(X) + \pi(X_1))/32$. Thus $\pi(X_1) \geq \pi(X)/15$.

Let $X_2 \subseteq X$ be the set of all nodes $x \in X$ such that x receives at least $\pi(x)/32$ units of flow in f_2. As before, we have $\pi(X_2) \geq \pi(X)/15$.

Now consider a subset $X' \subseteq X$ such that $\pi(X') \leq \pi(X)/15$. Note that $\pi(X') \leq \pi(X_1)$ and $\pi(X') \leq \pi(X_2)$. Consider the following multicommodity flow instance $\mathbf{d}$: $d(x', x) = \pi(x')\pi(x)/(32\pi(X_1))$ for each pair $(x', x) \in X' \times X_1$, $d(x, x') = \pi(x)\pi(x')/(32\pi(X_2))$ for each pair $(x, x') \in X_2 \times X'$, and $d(\cdot)$ is zero for all other pairs. Since $d(a, b) \leq \pi(a)\pi(b)/\pi(X)$ for all pairs of nodes (a, b), there is a feasible flow g that routes $\mathbf{d}$. The flow g satisfies the following properties:

- Each node $x \in X_1$ receives $\pi(x)\pi(X')/(32\pi(X_1)) \leq \pi(x)/32$ units of flow.
- Each node $x' \in X'$ sends $\pi(x')/32$ units of flow.
- Each node $x \in X_2$ sends $\pi(x)\pi(X')/(32\pi(X_2)) \leq \pi(x)/32$ units of flow.
- Each node $x' \in X'$ receives $\pi(x')/32$ units of flow.

By combining the flows f_1 and g, we get a congestion two flow from X' to Y in which each node in Y receives at most one unit of flow and each node $x' \in X'$ sends $\pi(x')/32$ units of flow. Similarly, by combining the flows f_2 and g, we get a congestion two flow from Y to X' in which each node in Y sends at most one unit of flow and each node $x' \in X'$ receives $\pi(x')/32$ units of flow. We scale down these flows by a factor of two to get feasible flows. $\square$

Second step: Finding a matching. In the second step, we use the set Y guaranteed by Theorem 4 in order to select a matching $\mathcal{M}' \subseteq \mathcal{M}$.

We will need the following theorem, which is a slight variant of Theorem 2.1 in [11]. Let G be a directed graph with integer arc capacities given by c. Let $s_1, s_2, \ldots, s_k$ be distinct source nodes and let t be a sink node. A non-negative vector $\mathbf{b} = (b_1, b_2, \ldots, b_k)$ is a *feasible* flow vector if there is a feasible flow in G in which each source s_i sends b_i units of flow to t and t receives $\sum_{i=1}^k b_i$ units of flow. Let $\mathcal{B}$ be the set of all feasible flow vectors. For a vector $\mathbf{b} \in \mathcal{B}$, let $F(\mathbf{b}) = \sum_{i=1}^k b_i$ denote the total flow and let $I(\mathbf{b})$ be the set of all indices i such that b_i is an integer.

Theorem 5 *Given* $\mathbf{b} \in \mathcal{B}$ *and* $j \notin I(\mathbf{b})$ *with* $b_j > 0$*, we can compute* $\mathbf{b}' \in \mathcal{B}$ *in polynomial time with* $b'_j = \lceil b_j \rceil$ *and* $F(\mathbf{b}') \geq F(\mathbf{b})$ *such that*

- $b'_i \leq b_i$ *for each* $i \in [k] - \{j\}$*, and*
- $b'_i = b_i$ *for each* $i \in I(\mathbf{b})$*.*

The difference between Theorem 5 and Theorem 2.1 in [11] is that the former theorem requires that $b'_i \leq b_i$ for each terminal $i \neq j$, whereas the latter theorem requires that $b'_i \leq \lceil b_i \rceil$. One can prove the theorem above using essentially the same argument as in [11].

Note that the flow augmentation theorem (Theorem 5) also applies to single-source networks and flows, since we can simply reverse the directions of all of the arcs. It also applies to node-capacitated routing using a standard reduction from node-capacitated directed networks to edge-capacitated directed networks.

Now we are ready to complete the proof of Theorem 3. Note that we may assume that $\pi(X) \geq c$, for some large enough constant c, since otherwise we can let $\mathcal{M}'$ be a single pair (u, v) of $\mathcal{M}$ that is routable in G. In the following, we assume that $\pi(X) \geq 2048$. Additionally, we may assume that $\pi(x) > 0$ for each node $x \in X$, since we can discard from X all the nodes x such that $\pi(x) = 0$.

Using Theorem 5, we can identify a large matching whose terminals can send one unit of flow to Y and receive one unit of flow from Y.

Lemma 4 *There is a matching* $\mathcal{M}' \subseteq \mathcal{M}$ *with the following properties. Let* X'_1 *be a set of nodes containing exactly one node from each pair in* $\mathcal{M}'$*, and let* $X'_2 = V(\mathcal{M}') - X'_1$ *be the partners of the nodes in* X'_1*. We have*

(C_1) $|\mathcal{M}'| = \Omega(|Y|)$.
(C_2) *There is a feasible single-commodity flow in* G *in which each node in* X'_1 *sends one unit of flow to* Y*.*
(C_3) *There is a feasible single-commodity flow in* G *in which each node in* X'_1 *receives one unit of flow from* Y*.*
(C_4) *There is a feasible single-commodity flow in* G *in which each node in* X'_2 *sends one unit of flow to* Y*.*
(C_5) *There is a feasible single-commodity flow in* G *in which each node in* X'_2 *receives one unit of flow from* Y*.*

Proof: Let $\mathcal{M}''$ be any subset of $\mathcal{M}$ such that $\pi(X)/16 \leq \pi(X'') \leq \pi(X)/15$, where X'' is the set of nodes participating in the pairs of $\mathcal{M}''$. Note that such a set $\mathcal{M}''$ exists, since $\pi(X) \geq 240$ and $\pi(x) \leq 1$ for each node $x \in X$. Let X''_1 be a set of nodes containing exactly one node from each pair in $\mathcal{M}''$, and let $X''_2 = V(\mathcal{M}'') - X''_1$ be the partners of the nodes in X''_1.

In the following, we use the flow augmentation theorem (Theorem 5) to select a matching $\mathcal{M}' \subseteq \mathcal{M}''$ with the desired properties.

We make four copies of G; let G_1, G_2, G_3, and G_4 denote the four copies of G. For each $i \in \{1, 3\}$, we construct a node-capacitated single-sink network H_i from G_i as follows. We start with $H_i = G_i$ and we assign a capacity of one to each node. We add to H_i a sink node t_i and a directed edge from each node in Y to t_i. For each $i \in \{2,4\}$, we construct a node-capacitated single-source network H_i from G_i as follows. We start with $H_i = G_i$ and we assign a capacity of one to each node. We add to H_i a source node s_i and a directed edge from s_i to each node in Y.

For each $i \in \{1,2,3,4\}$, we maintain a feasible flow vector $\mathbf{b}_i$ in H_i. If $i \in \{1,2\}$, $\mathbf{b}_i$ has an entry $\mathbf{b}_i(x)$ for each node $x \in X_1''$. If $i \in \{3,4\}$, $\mathbf{b}_i$ has an entry $\mathbf{b}_i(x)$ for each node $x \in X_2''$.

We initialize the flow vectors $\mathbf{b}_i$ as follows. Let f_1 be the flow from X'' to Y guaranteed by property (Q_1) (see the statement of Theorem 4). Let f_2 be the flow from Y to X'' guaranteed by property (Q_2). Note that, for each $i \in \{1,3\}$, f_1 translates to a flow in H_i from X'' to t_i; we let $\mathbf{b}_i(x)$ denote the amount of flow from x to t_i. Similarly, for each $i \in \{2,4\}$, f_2 translates to a flow in H_i from s_i to X''; we let $\mathbf{b}_i(x)$ denote the amount of flow from s_i to x.

Our goal is to use the flow augmentation theorem (Theorem 5) in order to select a matching $\mathcal{M}' \subseteq \mathcal{M}''$. The main idea behind the approach is the following. If we have a pair $(u,v) \in \mathcal{M}''$ whose flow is fractional (that is, $\mathbf{b}_1(u) = \mathbf{b}_2(u) = \mathbf{b}_3(v) = \mathbf{b}_4(v) \in (0,1)$), we use Theorem 5 in each copy G_i to increase the flow of u and v to 1. We repeatedly apply this procedure until the flow of each pair is either 0 or 1. The pairs with unit flow will give us the desired matching. We now describe this approach more formally.

If $\mathbf{b}$ is a flow vector on $Z \subseteq X''$, we let $F(\mathbf{b}) = \sum_{x \in Z} \mathbf{b}(x)$ be the total flow and $I(\mathbf{b})$ denote the set of all nodes $x \in Z$ such that $\mathbf{b}(x) = 1$. We will maintain flow vectors $\mathbf{b}_i$ that satisfy the following invariants:

- (I_1) For each $u \in X_1''$, we have $\mathbf{b}_1(u) = \mathbf{b}_2(u) = \mathbf{b}_3(v) = \mathbf{b}_4(v)$, where v is the partner of u.
- (I_2) For each $i \in \{1,2,3,4\}$, we have $F(\mathbf{b}_i) \geq (|Y|/256) - 4|I(\mathbf{b}_1)|$.

We can verify that the initial flow vectors satisfy the invariants as follows. For each pair $uv \in \mathcal{M}$, we have $\mathbf{b}_1(u) = \mathbf{b}_2(u) = \pi(u)/64$ and $\mathbf{b}_3(v) = \mathbf{b}_4(v) = \pi(v)/64$. Since $\pi(a) = \pi(b)$ for each pair $ab \in \mathcal{M}$, the flow vectors satisfy the first invariant. Note that $I(\mathbf{b}_i)$ is empty, since $\mathbf{b}_i(x) = \pi(x)/64 < 1$ for each $x \in X''$. Moreover, we have $F(\mathbf{b}_i) = \pi(X_1'')/64 = \pi(X'')/128$. Since $\pi(X'') \geq \pi(X)/16$ and $|Y| = \lfloor \pi(X)/8 \rfloor$, we have $F(\mathbf{b}_i) = \pi(X'')/128 \geq \pi(X)/2048 \geq |Y|/256$. Thus the flow vectors satisfy the second invariant.

Now consider flow vectors $\mathbf{b}_i$ that satisfy the invariants (I_1) and (I_2) above. Suppose that, for each $u \in X_1''$, we have $\mathbf{b}_1(u) \in \{0,1\}$. Let $X_1' = I(\mathbf{b}_1)$ and $X_2' = I(\mathbf{b}_3)$; by (I_1), X_2' is the set of all partners of the nodes of X_1'. Let $\mathcal{M}'$ be the set of all pairs $uv \in \mathcal{M}''$ such that $u \in X_1'$ and $v \in X_2'$. We can verify that $\mathcal{M}'$ is the desired matching as follows. We have $|X_1'| = F(\mathbf{b}_1) \geq (|Y|/256) - 4|X_1'|$ and thus $|X_1'| \geq |Y|/1280$. Thus X_1' and X_2' satisfy the conditions (C_1)-(C_5) in the theorem statement and we are done.

Therefore we may assume that there is a node $u \in X_1''$ such that $\mathbf{b}_1(u) \in (0,1)$. Let v be the partner of u. Recall that we have $\mathbf{b}_1(u) = \mathbf{b}_2(u) = \mathbf{b}_3(v) = \mathbf{b}_4(v)$. For each $i \in \{1,2\}$, we apply Theorem 5 with $G = H_i$, $\mathbf{b} = \mathbf{b}_i$, and $b_j = u$ in order to get a feasible flow vector $\mathbf{b}_i'$ such that $I(\mathbf{b}_i') \supseteq I(\mathbf{b}_i) \cup \{u\}$.

For each $i \in \{3,4\}$, we apply Theorem 5 with $G = H_i$, $\mathbf{b} = \mathbf{b}_i$, and $b_j = v$ in order to get a feasible flow vector $\mathbf{b}'_i$ such that $I(\mathbf{b}'_i) \supseteq I(\mathbf{b}_i) \cup \{v\}$. We construct flow vectors $\mathbf{b}''_i$ as follows. For each pair $zw \in \mathcal{M}''$, we let $\mathbf{b}''_1(z) = \mathbf{b}''_2(z) = \mathbf{b}''_3(w) = \mathbf{b}''_4(w) = \min\{\mathbf{b}'_1(z), \mathbf{b}'_2(z), \mathbf{b}'_3(w), \mathbf{b}'_4(w)\}$.

In the following, we show that the flows $\mathbf{b}''_i$ satisfy the second invariant. This follows from the properties guaranteed by Theorem 5. When we augment the flow of u to 1 in G_1 (or G_2), we decrease the total flow of all other pairs by at most 1. Similarly, when we augment the flow of v to 1 in G_3 (or G_4), we decrease the total flow of all other pairs by at most 1. Thus the total flow decrease in G_1, G_2, G_3, and G_4 is at most 4, and we charge this flow to the pair uv.

More formally, we claim that $F(\mathbf{b}''_i) \geq F(\mathbf{b}_i) - 4$ for each $i \in \{1,2,3,4\}$. Consider an index $i \in \{1,2\}$. Note that it suffices to show that $\sum_{z \in X''_1 - \{u\}} \Big(\mathbf{b}_i(z) - \mathbf{b}''_i(z)\Big) \leq 4$. We have

$$
\begin{aligned}
&\sum_{z \in X''_1 - \{u\}} \Big(\mathbf{b}_i(z) - \mathbf{b}''_i(z)\Big) = \sum_{zw \in \mathcal{M}'' - \{uv\}} \Big(\mathbf{b}_i(z) - \min\{\mathbf{b}'_1(z), \mathbf{b}'_2(z), \mathbf{b}'_3(w), \mathbf{b}'_4(w)\}\Big) \\
&= \sum_{zw \in \mathcal{M}'' - \{uv\}} \max\{\mathbf{b}_1(z) - \mathbf{b}'_1(z), \mathbf{b}_2(z) - \mathbf{b}'_2(z), \mathbf{b}_3(w) - \mathbf{b}'_3(w), \mathbf{b}_4(w) - \mathbf{b}'_4(w)\} \\
&\qquad \Big(\text{Since } \mathbf{b}_1(z) = \mathbf{b}_2(z) = \mathbf{b}_3(w) = \mathbf{b}_4(w)\Big) \\
&\leq \sum_{zw \in \mathcal{M}'' - \{uv\}} (\mathbf{b}_1(z) - \mathbf{b}'_1(z) + \mathbf{b}_2(z) - \mathbf{b}'_2(z) + \mathbf{b}_3(w) - \mathbf{b}'_3(w) + \mathbf{b}_4(w) - \mathbf{b}'_4(w)) \\
&\qquad \Big(\text{The terms in the max are non-negative by the first bullet in Theorem 5}\Big) \\
&= \sum_{i=1}^{4} \Big(F(\mathbf{b}_i) - F(\mathbf{b}'_i)\Big) + \sum_{i=1}^{4} \Big(\mathbf{b}_i(u) - \mathbf{b}'_i(u)\Big) + \sum_{i=1}^{4} \Big(\mathbf{b}_i(v) - \mathbf{b}'_i(v)\Big) \\
&\leq 4
\end{aligned}
$$

The last inequality follows from the fact that $F(\mathbf{b}_i) \leq F(\mathbf{b}'_i)$ for all $i \in \{1,2,3,4\}$, $\mathbf{b}_i(u) - \mathbf{b}'_i(u)$ is at most 0 if $i \in \{1,2\}$ and at most 1 if $i \in \{3,4\}$, $\mathbf{b}_i(v) - \mathbf{b}'_i(v)$ is at most 1 if $i \in \{1,2\}$ and at most 0 if $i \in \{3,4\}$.

A very similar argument shows that, for each $i \in \{3,4\}$, we have $F(\mathbf{b}''_i) \geq F(\mathbf{b}_i) - 4$. Thus we have $F(\mathbf{b}''_i) \geq F(\mathbf{b}_i) - 4 \geq (|Y|/256) - 4(|I(\mathbf{b}_1)| + 1)$ for each $i \in \{1,2,3,4\}$. Since $|I(\mathbf{b}''_1)| \geq |I(\mathbf{b}_1)| + 1$, we have $F(\mathbf{b}''_i) \geq (|Y|/256) - 4|I(\mathbf{b}''_1)|$ and thus the second invariant is also satisfied.

We repeatedly apply the flow augmentation procedure until the flow of each pair is either zero or one. The pairs with unit flow will give us the desired matching $\mathcal{M}'$. □

Let $\mathcal{M}'$ be the set of pairs guaranteed by Lemma 4 and let X' be the set of terminals participating in the pairs of $\mathcal{M}'$. We can show that X' is 1/32-flow-well-linked as follows. Note that the properties $(C_2) - (C_5)$ gives us the following flows: a congestion two flow from X' to Y in which each node in

X' sends one unit of flow and each node in Y receives at most two units of flow, and a congestion two flow from Y to X' in which each node in Y sends at most two units of flow and each node in X' receives one unit of flow. We scale these flows by a factor of 8 to ensure that each node in Y sends and receives at most 1/4 units of flow. Since Y is 1/4-flow-well-linked, it follows from Lemma 3 that X' is 1/32-flow-well-linked; here we applied the lemma with $A = Y$, $\pi(y) = 1/4$ for each $y \in Y$, $B = X'$, $\pi'(x') = 1/8$ for each $x' \in X'$. This completes the proof of Theorem 3.

References

1. I. Adler. Directed tree-width examples. *Journal of Combinatorial Theory, Series B*, 97(5):718 – 725, 2007.
2. M. Andrews, J. Chuzhoy, V. Guruswami, S. Khanna, K. Talwar, and L. Zhang. Inapproximability of edge-disjoint paths and low congestion routing on undirected graphs. *Combinatorica*, 30(5):485–520, 2010.
3. M. Andrews, J. Chuzhoy, S. Khanna, and L. Zhang. Hardness of the undirected edge-disjoint paths problem with congestion. In *Proc. of IEEE FOCS*, pages 226–241, 2005.
4. C. Chekuri and J. Chuzhoy. Large-treewidth graph decompositions and applications. In *Proc. of ACM STOC*, 2013.
5. C. Chekuri and J. Chuzhoy. Polynomial bounds for the grid-minor theorem. In *Proc. of ACM STOC*, 2014.
6. C. Chekuri and A. Ene. Poly-logarithmic approximation for maximum node disjoint paths with constant congestion. In *Proc. of ACM-SIAM SODA*, 2013.
7. C. Chekuri, S. Kannan, A. Raja, and P. Viswanath. Multicommodity flows and cuts in polymatroidal networks. In *Proc. of ITCS*, pages 399–408, 2012.
8. C. Chekuri, S. Khanna, and F. B. Shepherd. The all-or-nothing multicommodity flow problem. *SIAM Journal on Computing*, 42(4):1467–1493, 2013.
9. C. Chekuri, S. Khanna, and F.B. Shepherd. Multicommodity flow, well-linked terminals, and routing problems. In *Proc. of ACM STOC*, pages 183–192, 2005.
10. C. Chekuri, S. Khanna, and F.B. Shepherd. Well-linked terminals for node-capacitated routing problems. Manuscript, 2005.
11. C. Chekuri, S. Khanna, and F.B. Shepherd. An $O(\sqrt{n})$ approximation and integrality gap for disjoint paths and unsplittable flow. *Theory of Computing*, 2(7):137–146, 2006.
12. C. Chekuri, M. Mydlarz, and F.B. Shepherd. Multicommodity demand flow in a tree and packing integer programs. *ACM Transactions on Algorithms*, 3(3):27, 2007.
13. J. Chuzhoy. Routing in undirected graphs with constant congestion. *ArXiv preprint ArXiv:1107.2554*, 2011. Extended abstract in STOC 2012.
14. J. Chuzhoy, V. Guruswami, S. Khanna, and K. Talwar. Hardness of routing with congestion in directed graphs. In *Proc. of ACM STOC*, pages 165–178, 2007.
15. J. Chuzhoy and S. Khanna. Polynomial flow-cut gaps and hardness of directed cut problems. *Journal of the ACM*, 56(2):6, 2009.
16. J. Chuzhoy and S. Li. A polylogarithimic approximation algorithm for edge-disjoint paths with congestion 2. In *Proc. of IEEE FOCS*, 2012.
17. S. Even, A. Itai, and A. Shamir. On the complexity of timetable and multicommodity flow problems. *SIAM Journal on Computing*, 5(4):691–703, 1976.
18. U. Feige, M.T. Hajiaghayi, and J.R. Lee. Improved approximation algorithms for minimum weight vertex separators. *SIAM Journal on Computing*, 38:629–657, 2008.
19. S. Fortune, J. Hopcroft, and J. Wyllie. The directed subgraph homeomorphism problem. *Theoretical Computer Science*, 10(2):111–121, 1980.
20. N. Garg, V.V. Vazirani, and M. Yannakakis. Primal-dual approximation algorithms for integral flow and multicut in trees. *Algorithmica*, 18(1):3–20, 1997.
21. T. Johnson, N. Robertson, P. D. Seymour, and R. Thomas. Directed tree-width. *Journal of Combinatorial Theory, Series B*, 82(1):138–154, 2001.

22. R. M. Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations*, pages 85–103, 1972.
23. K. Kawarabayashi and S. Kreutzer. An excluded grid theorem for digraphs with forbidden minors. In *Proc. of ACM-SIAM SODA*, 2014.
24. Yusuke Kobayashi Ken-ichi Kawarabayashi and Stephan Kreutzer. An excluded half-integral grid theorem for digraphs and the directed disjoint paths problem. In *Proc. of ACM STOC*, 2014.
25. P. N. Klein, S. A. Plotkin, and S. Rao. Excluded minors, network decomposition, and multicommodity flow. In *Proc. of ACM STOC*, pages 682–690, 1993.
26. P. N. Klein, S. A. Plotkin, S. Rao, and E. Tardos. Approximation algorithms for Steiner and directed multicuts. *Journal of Algorithms*, 22(2):241–269, 1997.
27. T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM*, 46(6):787–832, 1999.
28. N. Linial, E. London, and Y. Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995.
29. B. Reed. Introducing directed tree width. *Electronic Notes in Discrete Mathematics*, 3:222–229, 1999.
30. Michael E. Saks, Alex Samorodnitsky, and Leonid Zosin. A lower bound on the integrality gap for minimum multicut in directed networks. *Combinatorica*, 24(3):525–530, 2004.
31. A. Srinivasan. Improved approximations for edge-disjoint paths, unsplittable flow, and related routing problems. In *Proc. of IEEE FOCS*, pages 416–425, 1997.