Approximate Proof-Labeling Schemes

Keren Censor-Hillel[1], Ami Paz[1], and Mor Perry[2]

[1] Department of Computer Science, Technion, Israel. ‡
{ckeren,amipaz}@cs.technion.ac.il.
[2] Department of Electrical Engineering, Tel Aviv University, Israel. §
mor@eng.tau.ac.il.

Abstract. We study a new model of verification of boolean predicates over distributed networks. Given a network configuration, the proof-labeling scheme (PLS) model defines a distributed proof in the form of a label that is given to each node, and all nodes locally verify that the network configuration satisfies the desired boolean predicate by exchanging labels with their neighbors. The *proof size* of the scheme is defined to be the maximum size of a label.

In this work, we extend this model by defining the *approximate proof-labeling scheme* (APLS) model. In this new model, the predicates for verification are of the form $\psi \leq \varphi$, where $\psi, \varphi : \mathcal{F} \to \mathbb{N}$ for a family of configurations $\mathcal{F}$. Informally, the predicates considered in this model are a comparison between two values of the configuration. As in the PLS model, nodes exchange labels in order to locally verify the predicate, and all must accept if the network satisfies the predicate. The soundness condition is relaxed with an approximation ration α, so that only if $\psi > \alpha\varphi$ some node must reject. We show that in the APLS model, the proof size can be much smaller than the proof size of the same predicate in the PLS model. Moreover, we prove that there is a tradeoff between the approximation ratio and the proof size.

Keywords: Distributed graph algorithms, Distributed verification, Approximation algorithms, Primal-dual algorithms

1 Introduction

1.1 Context and Objective

Verification of a given property in decentralized systems finds applications in various domains, such as, checking the result obtained from the execution of a distributed program [5, 20], establishing lower bounds on the time required for distributed approximation [11], estimating the complexity of logic required for distributed run-time verification [21], general distributed complexity theory [19], and the construction of self stabilizing algorithms [8, 26].

In the distributed setting, a network configuration G_s is represented by an underlying graph and a state assignment. The underlying graph has nodes and edges, where each node represents a processor and each edge represents a communication link between two processors. The state assignment is the state of each node, which can contain a unique identifier, edge weights, a specified subset of incident edges, an output of a distributed algorithm and more. In order to verify that a network configuration has a specified property, nodes exchange messages along the edges and output either TRUE or FALSE depending on whether the local configuration is consistent with a legal state of the network. The distributed verification process is correct if all nodes return TRUE on legal configurations, and on every illegal configuration at least one node returns FALSE. Some properties are local by nature and easy to verify, for example, whether a specified subset of edges is a matching in the graph. However, many other properties cannot be verified in less than diameter time, even if message size and local computational power are unbounded, for example, whether a specified matching is of maximum cardinality.

In order to cope with strong time lower bounds, Korman, Kutten, and Peleg [27] have introduced a computational model, called *proof-labeling schemes* (PLSs), where nodes are given auxiliary global information in the form of *labels*. A proof-labeling scheme for a predicate $\mathcal{P}$ consists of a *prover* and a *verifier*. For every legal state of the network, the prover assigns a *label* to every node. The verifier is a distributed algorithm, which uses one round of communication to exchange labels between neighbors and output TRUE or FALSE at each node, as a function of the state and label of the node and the labels it receives from its neighbors. A PLS satisfies *completeness* if for every legal configuration, with the labels

‡Supported by ISF individual research grant 1696/14.

§Partially supported by Apple Graduate Fellowship.

Approximation Ratio	Upper Bound		Lower Bound	
Exact	$O(n \log n)$	(Section 3)	$\Omega(n/k)$	(Theorem 1)
$3/2 - \epsilon$			$\Omega\left(n/\log^2 n\right)$	(Theorem 3)
$3/2$	$O\left(\sqrt{n}\log^2 n\right)$	(Theorem 2)		
2	$O(\log n)$	(Theorem 4)		

Table 1. APLS for $(D \leq k)$ on general graphs—upper and lower bounds on proof size.

assigned by the prover, all nodes output TRUE, and it satisfies *soundness* if for every illegal configuration and every label assignment, some node outputs FALSE.

The complexity measure used in evaluating the quality of a PLS is the maximum size of a label, which is called the *proof size*. It is known that, for every sequentially decidable graph property, there exists a PLS with proof size $O(m \log n)$ where n is the number of nodes and m is the number of edges in the network [6,22,27]. For some properties, lower bounds on the proof size have been proven in this model, for example $\Omega(\log n)$ for verification of a spanning-tree [27] and bi-connectivity [6], $\Omega(n^2/\log n)$ for verifying that the graph is not 3-colorable [22], and $\Omega(\log^2 n)$ for verification of a minimum-weight spanning-tree [25], assuming that the maximal edge-weight W satisfies $\log n < W \leq n^c$ for some constant c.

As in the computational framework, variations of the model may allow us to break known lower bounds. It has been suggested to use super-constant number of rounds in verification [7,26]. In the former, a linear reduction of proof size is proven for acyclicity and the universal scheme. In the latter, they present a scheme for minimum-weight spanning-tree with $O(\log n)$ proof size and $O(\log^2 n)$ rounds. In [6] it was suggested to distinguish between labels and communication in the verification process, and to use randomization in order to reduce the communication complexity of verification. They show an exponential reduction in the communication complexity of every scheme at the cost of increasing the proof size by a factor of the maximum degree.

Yet, some properties are still harder. In Section 3 we show that any PLS for $D \leq k$ must have labels of $\Omega(n)$ bits, where D is the diameter of the graph and $k \in \mathbb{N}$ is a constant. A natural way to circumvent this lower bound is through approximation, e.g., by defining a 2-approximation for the problem by the predicate $D \leq 2k$, and hoping for smaller proof size. However, this approach is bound to fail: any PLS for $D \leq 2k$ is also a PLS for $D \leq k'$, for $k' = 2k$, so the same lower bound holds for this definition of approximation.

Inspired by the above example, we present and investigate a new concept of *approximate* proof-labeling schemes (APLSs for short) for optimization problems. Let $\psi, \varphi : \mathcal{F} \to \mathbb{N}$ be two functions from a family of configurations to the natural numbers. Assume that we are interested in verifying for every $G_s \in \mathcal{F}$ whether $\psi(G_s) \leq \varphi(G_s)$, and let $\alpha > 1$ be the approximation ratio. If $\psi(G_s) \leq \varphi(G_s)$ then there is an assignment of labels such that all nodes output TRUE, and if $\psi(G_s) > \alpha\varphi(G_s)$ then for every label assignment at least one node outputs FALSE. If $\varphi(G_s) < \psi(G_s) \leq \alpha\varphi(G_s)$, we do not have any promise. Put differently, we are promised that if all nodes output TRUE, then $\psi(G_s) \leq \alpha\varphi(G_s)$, i.e., the approximation holds. This concept indeed allows us to find schemes with shorter labels: we show a 2-APLS for $D \leq k$ with proof size of only $O(\log n)$ bits, and a 3/2-APLS for $D \leq k$ with proof size of $O(\sqrt{n}\log^2 n)$ bits.

1.2 Our Contribution

In this paper we introduce and formalize the concept of *approximate* proof-labeling schemes. We study the complexity of verification of two fundamental problems in this model: diameter and maximum weight matching. We start by considering the verification of a specified upper bound k on the network diameter D (see summary of results in Table 1), and show that for every $k = k(n)$, the proof size of any PLS for $D \leq k$ is $\Omega(n/k)$. In the APLS model, as outlined above, we present a 3/2-APLS for $D \leq k$ with $O(\sqrt{n}\log^2 n)$ proof size, and prove that we cannot obtain a better approximation ratio with the same asymptotic proof size. Specifically, we prove that for every k there exists an $\epsilon \in \Theta(1/k)$ such that the proof size of any $(3/2 - \epsilon)$-APLS for $D \leq k$ is $\Omega(n/\log^2 n)$. Then, we turn to show that if we increase the approximation ratio we can construct an even more efficient scheme. In particular, we show a simple 2-APLS for $D \leq k$ with proof size $O(\log n)$. To our knowledge, the problem of verifying an upper bound on the diameter in general graphs has not been studied before in the context of PLSs.

The second property we consider is verifying that a specified matching M have the maximum possible weight (see summary of results in Table 2). For this property we are interested in bounding from below the weight of the matching w.r.t. the weight of the maximum matching $w(\text{MWM})$. We present a 2-APLS for

Approximation Ratio	Graph Family	Upper Bound		Lower Bound	
Exact	Paths	$O(\log n + \log W)$	[27]		
Exact	Bipartite	$O(\log W)$	[22]		
2	Trees	$O(\log n + \log W)$	[27]		
2	Any graph	$O(\log W)$	(Theorem 6)		
Any	Any graph			$\Omega(1)$	(Section 4)

Table 2. APLS for $(w(M) \geq w(\mathrm{MWM}))$—upper and lower bounds on proof size.

$w(M) \geq w(\mathrm{MWM})$ with $O(\log W)$ proof size, where W is the maximum edge-weight in the network. This improves upon a previous result presented in [27], with $O(\log n + \log W)$ proof size for a 2-approximation of the maximum weight matching on trees. We note that the notion of approximation in [27] is different from our definition: they argue that there exists a subset of 2-approximated configurations that the scheme verifies, but do not promise that any configuration with an optimal matching is verified successfully.

We use various techniques to obtain our results. The lower bounds for proof complexity are achieved using reductions for nondeterministic communication complexity [22], a lower bound graph presented in [23] and a recent constructions of [1]. The design of the APLSs is based on approximation algorithms for the diameter problem [2], and on complementary slackness conditions for primal-dual problems.

1.3 Related Work

Approximation algorithms were studied extensively in both sequential and distributed computing. In the sequential model, unless P = NP, there are no polynomial-time algorithms for NP-hard problems, and thus efficient approximation algorithms for the related optimization problems are widely studied [31]. Moreover, even for problems for which polynomial time algorithms exist, there is sometimes a need for faster algorithms that give an approximate solution.

One example is the problem of determining the diameter of a graph. While the problem is solvable in polynomial time, faster approximation algorithms are studied. A trivial 2-approximation algorithm in unweighted graphs goes through building a single BFS tree in $O(n + m)$ time, and measuring its depth. An $\tilde{O}(m\sqrt{n} + n^2)$ time 3/2-approximation algorithm for the diameter was presented in [2], and was later improved in [30] to $\tilde{O}(m\sqrt{n})$ time algorithms using randomization. A deterministic improvement to [2] was presented in [9]. Distributed algorithms for computing the diameter were presented in [23] and [29], and both also provide approximation algorithms for the problem. Lower bounds on computing and approximating the diameter in the CONGEST model were presented in [1, 24].

Distributed decision and verification schemes deal with verifying that a given instance satisfies some given boolean predicate. Distributed decision and verification has been formalized in various models to suit its myriad applications. These models include proof-labeling schemes (PLSs) [27], locally checkable proofs (LCP) [22], and several complexity classes [19]. The complexity classes presented in the latter include LD—local decision—which includes all properties that can be decided using a constant number of rounds and no additional information, and NLD—non-deterministic local decision—which includes all properties that can be decided in a constant number of rounds with additional information in the form of a certificate given to each node. While NLD and PLS are closely related, they differ in that NLD certificates are independent of node identifiers. Since PLS labels may depend on node identifiers, there is a PLS for every sequentially decidable property on ID based networks, while not all sequentially decidable properties are in NLD. For more details, we refer the reader to a survey of this field of research [12].

The concept of PLS was introduced by Korman, Kutten, and Peleg in [27]. Among other results, they show a $\Theta(\log n)$ bound on the proof size of the diameter of trees, and the same bound also for the proof size of a lower bound on the diameter in general graphs. In addition, they present two $O(\log n + \log W)$ schemes to verify a maximum weight matching: one on paths, and the other is a 2-approximation of maximum weight matching on trees.

Since the introduction of proof-labeling schemes, many aspects of this mechanism were studied. Schemes where nodes may communicate to a constant distance that is greater than 1 were studied in [22]. For the maximum cardinality matching problem, they show that the proof size on the family of bipartite graphs is $\Theta(1)$, and on the family of cycle graphs is $\Theta(\log n)$. For maximum weight matching, they present a scheme for the family of bipartite graphs, with $O(\log W)$ proof size, using techniques similar to the ones we use. Moreover, [22] was the first to use nondeterministic communication complexity lower bounds in order to achieve lower bounds on the verification complexity of a PLS.

Schemes with super-constant verification time were presented in [26]. Verification processes in which the global result is not restricted to be the conjunction of local outputs had been studied in [3,4]. The role of unique node identifiers in local decision and verification was extensively studied in [16–18]. The use of randomization in verification process in order to reduce communication was presented in [6]. Proof-labeling schemes in directed networks were studied in [14], where both one-way and two-way communication over directed edges had been considered. Verification schemes for dynamic networks, where edges may appear or disappear after label assignment and before verification, were studied in [15]. Finally, a hierarchy of local decision as an interaction between a prover and a disprover was presented in [13].

2 Model and Definitions

2.1 Computational Framework

A network is modeled by a connected, undirected, simple graph $G = (V, E)$, with $|V| = n$ nodes and $|E| = m$ edges. Each node represents a processor, and each edge represents a communication link. We do not assume the a processor initially knows to which other processors it is connected, but only that its communication links are enumerated by *port numbers*. A *configuration* G_s is graph $G = (V, E)$ along with a state assignment function $s : V \to S$, where S is called the *state space*. The state $s(v)$ of a node v includes all local input to v. In particular, the state includes port numbers of adjacent edges, the node's identity (if the network is not anonymous) or other data, e.g., the result of an algorithm. We sometimes consider *weighted networks*, in which the graph is accompanied with an edge weight function $w : V \to \{1, \dots, W\}$, in which case the state of a node includes the weights of its adjacent edges.[3]

In this work, we always assume non-anonymous networks, i.e., every node v is provided with a unique identity $\mathrm{ID}(v)$, which is part of the state of v.

2.2 Proof-Labeling Schemes

Given a family $\mathcal{F}$ of network configurations and a boolean predicate $\mathcal{P}$ over $\mathcal{F}$, a *proof-labeling scheme* (*PLS*) for $(\mathcal{F}, \mathcal{P})$ is a mechanism for deciding $\mathcal{P}(G_s)$ for every $G_s \in \mathcal{F}$. A PLS consists of two components: a *prover* $\mathbf{p}$, and a *verifier* $\mathbf{v}$. Given any legal configuration $G_s \in \mathcal{F}$ (i.e., a configuration satisfying $\mathcal{P}$), the prover assigns a bit string $\ell(v)$ to every node v, called the *label* of v. The verifier is a local distributed algorithm running concurrently at every node. At each node v, it takes as input the state $s(v)$ of v, its label $\ell(v)$ and the labels of all its neighbors, i.e., the list $(\ell(v_1) \dots \ell(v_d))$, where d is the degree of v, and v_i is the neighbor of v reachable from port number i. The outputs of the verifier at each node is a boolean value. If the outputs are `TRUE` at all nodes, $\mathbf{v}$ is said to *accept* the configuration, and otherwise (i.e., $\mathbf{v}$ outputs `FALSE` in at least one node) $\mathbf{v}$ is said to *reject* the configuration. For correctness, a PLS $(\mathbf{p}, \mathbf{v})$ for $(\mathcal{F}, \mathcal{P})$ must satisfy the following requirements, for every $G_s \in \mathcal{F}$:

- If $\mathcal{P}(G_s) =$ `TRUE` then, using the labels assigned by $\mathbf{p}$, the verifier $\mathbf{v}$ accepts G_s.
- If $\mathcal{P}(G_s) =$ `FALSE` then, for every label assignment, the verifier $\mathbf{v}$ rejects G_s.

The *proof size* of a PLS $(\mathbf{p}, \mathbf{v})$ is the maximum length of a label assigned by the prover $\mathbf{p}$ on a legal configuration $G_s \in \mathcal{F}$.

2.3 The New Model: Approximate Proof-Labeling Schemes

In this paper we focus on predicates that represent minimization or maximization problems. Formally, we are given two functions $\psi, \varphi : \mathcal{F} \to \mathbb{N}$, and we are interested in the predicate $\psi(G_s) \leq \varphi(G_s)$. Note that ψ or φ may be constant, e.g., in verifying an upper bound on the diameter of the graph, one can be interested in verifying $D(G_s) \leq k$. In some cases, classic verification might be too expansive, as proven in Section 3, and so we extend the definition of PLSs to *approximate proof-labeling schemes* (APLSs). We relax the requirements of a PLS so that a configuration for which the inequality $\psi(G_s) \leq \varphi(G_s)$ holds is guaranteed to be accepted by the scheme, while a configuration for which $\psi(G_s)$ much larger than $\varphi(G_s)$ is guaranteed to be rejected. Formally, for $\alpha \geq 1$, an α-APLS $(\mathbf{p}, \mathbf{v})$ for $(\mathcal{F}, (\psi \leq \varphi))$ must satisfy the following requirements, for every $G_s \in \mathcal{F}$:

- If $\psi(G_s) \leq \varphi(G_s)$ then, using the labels assigned by $\mathbf{p}$, the verifier $\mathbf{v}$ accepts G_s.
- If $\psi(G_s) > \alpha\varphi(G_s)$ then, for every label assignment, the verifier $\mathbf{v}$ rejects G_s.

[3] Recall that W is the maximum weight of an edge in the graph. If $W = 1$, we interpret $O(\log W)$ as $O(1)$.

The *proof size* of an APLS is defined similarly to that of a PLS. Our definitions naturally extend to predicates of the form $\psi \geq \varphi$, $\psi < \varphi$ and $\psi > \varphi$.

Finally, we note that although the definition of an APLS might seem to resemble definitions from the field of property testing, they are inherently different. Our measure for how close a graph is to satisfy a property is entirely algebraic, and has nothing to do with changing the graph by adding or removing edges. Moreover, all schemes presented in this paper are deterministic.

2.4 Problem Definitions

Diameter. Given a configuration G_s with an underlying graph $G = (V, E)$ and an edge weight function w, for every two nodes $u, v \in V$ denote by $\text{dist}(u, v)$ the length of the shortest (unweighted) path between u and v in G_s, and by $\text{dist}_w(u, v)$ the minimum weight of a path between u and v in G_s. The *unweighted diameter* of G_s, denoted by $D(G_s)$, is defined as $\max\{\text{dist}(u, v) \mid u, v \in V\}$. Similarly, The *weighted diameter* of G_s, denoted by $D_w(G_s)$, is defined as $\max\{\text{dist}_w(u, v) \mid u, v \in V\}$.

The first set of problems we consider in this work are problems of bounding the weighted and unweighted diameters from above.

Definition 1. *Let $\mathcal{F}$ be the family of all weighted connected undirected configurations and let $G_s \in \mathcal{F}$. For every integer $k = k(n)$, we define the problems $(\mathcal{F}, (D \leq k))$ and $(\mathcal{F}, (D_w \leq k))$.*

A *breadth-first search* (BFS) tree in a weighted or unweighted graph G_s from a root $r \in V$ is a tree consisting of a shortest (unweighted) path from r to every node in V. If the graph is weighted, we are also interested in a *shortest weighted distance tree* consisting of a shortest weighted path from a root node r to every node in V. Throughout the paper, we use known schemes for verification of a BFS tree and a shortest weighted distance tree [27]. They prove that for the verification of these trees it is enough to give every node the identity of the root and the distance from the root. Therefore, proof size is $O(\log n)$ for a BFS tree and $O(\log n + \log W)$ for a shortest weighted distance tree.

Matchings. Given a configuration G_s with an underlying graph $G = (V, E)$, an edge weight function w, and an edge subset $M \subset E$, M is a *matching* in G if no two edges in M share a node. The weight of a matching M, denoted by $w(M)$, is the sum of weights of all edges in M. We say that a matching M is a maximum weight matching (MWM) if $w(M) \geq w(M')$ for every matching M' in G.

Another problem we consider, of a different flavor, is to verify that a specified matching is a maximum weight matching.

Definition 2. *Let $\mathcal{F}_M$ be the family of all weighted connected undirected configurations with a specified matching M. Let $G_s \in \mathcal{F}$ and let* MWM *be a maximum weight matching in G_s. We define the problem $(\mathcal{F}_M, (w(M) \geq w(\text{MWM})))$.*

Note that although $w(M) > w(\text{MWM})$ is not possible (since M is promised to be a matching), the problem is defined to follow the structure of APLSs.

2.5 Two-Party Communication Complexity

Given two vectors $x, y \in \{0, 1\}^s$, we say the vectors are not disjoint, and write $\text{DISJ}(x, y)$=`FALSE`, if there exists an index $i \in [s]$ such that $x_i = y_i = 1$. Otherwise, the vectors are disjoint, and $\text{DISJ}(x, y)$=`TRUE`. In the Set-Disjointness two-party communication problem, two players denoted Alice and Bob are given two vectors, $x, y \in \{0, 1\}^s$ respectively, and they need to decide whether $\text{DISJ}(x, y)$=`TRUE` or $\text{DISJ}(x, y)$=`FALSE`. (See [28] for complete definitions and discussion.)

Given their inputs, the players communicate by a deterministic *protocol*, and eventually output $\text{DISJ}(x, y)$=`TRUE` or $\text{DISJ}(x, y)$=`FALSE`. A well known result in communication complexity asserts that in any protocol, Alice and Bob must exchange $\Omega(s)$ bits in order to correctly determine the value of $\text{DISJ}(x, y)$.[4]

In the nondeterministic case of the problem, Alice and Bob use auxiliary bit strings, which each of them nondeterministically chooses, and then run a deterministic protocol in order to determine the value of $\text{DISJ}(x, y)$. We are interested in the best assignment of auxiliary strings, i.e. the one that allows the players to minimize the number of bits exchanged. For example, if $\text{DISJ}(x, y)$=`FALSE` and Alice and Bob both use the index i such that $x_i = y_i = 1$ as an auxiliary string, then they only need to exchange $O(\log s)$, to verify they have the same index. On the other hand, a celebrated result [28] asserts that when $\text{DISJ}(x, y)$=`TRUE`, Alice and Bob must communicate $\Omega(s)$ even with an optimal assignment of auxiliary strings, i.e. nondeterminism cannot help Alice and Bob in asymptotically minimizing the communication.

[4] This lower bound holds also for randomized protocols, which we do not discuss in this work.

3 PLS and APLS for Diameter

Verifying that the diameter of the graph is bounded from above by a specified value can be done by a PLS with $O(n \log n)$ proof size (and $O(n(\log n + \log W))$ for weighted diameter). Simply construct a BFS tree (respectively, a shortest weighted distance tree) from every node, verify it and locally verify at each node that all of its distances are bounded by the specified value. We now show that in the PLS model, for a constant bound k, the proof size cannot be improved by more than a $\Theta(\log n)$ factor, i.e., it must have an $\Omega(n)$ proof size. Moreover, for every $k = k(n)$, we show a lower bound for the PLS proof size.

Let $s = \frac{n}{k} - 1$, and assume s is an even integer. Consider the following graph family $\{G_{x,y}\}$ over the following set of n nodes. Let $A_1 = a_1, ..., a_{s/2}$, $A_2 = a_{s/2+1}, ..., a_s$, $B_1 = b_1, ..., b_{s/2}$, and $B_2 = b_{s/2+1}, ..., b_s$ be four cliques, where each a_i is connected to b_i with a path of length $k-1$, consisting of a_i, b_i, and $k-2$ new nodes unique to this path. An additional node a is connected to every a_i by an edge, an additional node b is connected to every b_i by an edge, and there is a $(k-1)$-node path connecting a and b with another new $k-2$ nodes. Given an instance (x, y) of the Set-Disjointness problem over $(s/2)^2$ elements, enumerate Alice's input as x_{ij} with $i \in \{1, \dots, s/2\}$ and $j \in \{s/2+1, \dots, s\}$, and similarly for Bob's input, y_{ij}. To complete the construction of $G_{x,y}$, add an edge (a_i, a_j) if and only if $x_{ij} = 0$, and we add an edge (b_i, b_j) if and only if $y_{ij} = 0$.

If $\frac{n}{k} - 1$ is not an even integer, we choose s to be the largest even integer such that $s < \frac{n}{k} - 1$, add nodes to described construction to complement the number of nodes to n, and connect all additional nodes to all neighbors of b.

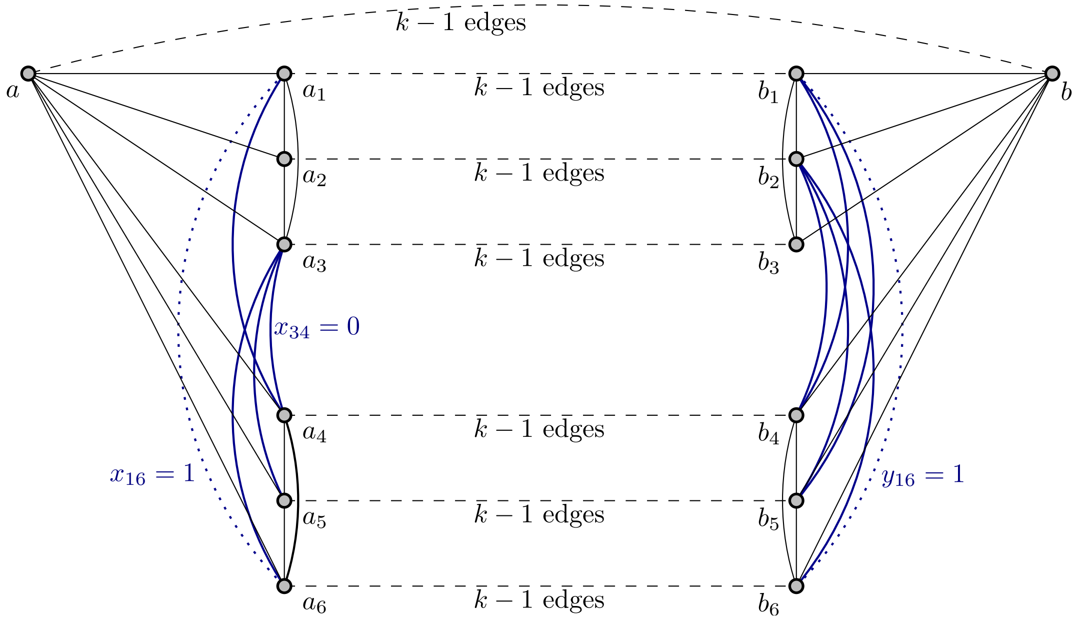

Fig. 1. The diameter lower bound construction for $s = 6$.
Here, $x = \begin{pmatrix} 0&1&1 \\ 1&1&1 \\ 0&0&0 \end{pmatrix}$ and $y = \begin{pmatrix} 0&0&1 \\ 0&0&0 \\ 1&1&1 \end{pmatrix}$, where the matrix rows are indexed by $\{1, 2, 3\}$ and the columns by $\{4, 5, 6\}$.
Since $x_{16} = y_{16} = 1$, the dotted edges are missing and the distance between a_1 and b_6 is greater than k.

Lemma 1. *$D(G_{x,y}) \leq k$ if and only if* $\text{DISJ}(x, y) = $ *TRUE.*

Proof. If $\text{DISJ}(x, y) = \texttt{TRUE}$, then for each $\{i, j\}$, at least one of the edges (a_i, a_j) or (b_i, b_j) exists in $G_{x,y}$. Let u and v be any two nodes in $G_{x,y}$. Suppose that u is on the path $(a_i \rightsquigarrow b_i)$ and v is on the path $(a_j \rightsquigarrow b_j)$, where $i, j \in \{1, \dots, s/2\}$. If $i = j$, clearly, $\text{dist}(u, v) \leq k - 1$. Otherwise, by assumption, either the cycle $(a \to a_i \rightsquigarrow b_i \to b_j \rightsquigarrow a_j \to a)$ or the cycle $(b \to b_j \rightsquigarrow a_j \to a_i \rightsquigarrow b_i \to b)$ exists and its length is $2k + 1$. Hence, every two nodes in the cycle are at distance at most k from each other, and $\text{dist}(u, v) \leq k$. Suppose now that either u or v is on the path $(a \rightsquigarrow b)$ and the other node is on the path $(a_i \rightsquigarrow b_i)$, $i \in \{1, \dots, s\}$. The length of the cycle $(a \to a_i \rightsquigarrow b_i \to b \rightsquigarrow a)$ is $2k$, and since u and v are on this cycle, $\text{dist}(u, v) \leq k$. Finally, if both u and v are on the path $(a \rightsquigarrow b)$, clearly, $\text{dist}(u, v) \leq k - 1$, and we conclude that $D(G_{x,y}) \leq k$.

If $\mathrm{DISJ}(x,y) = \texttt{FALSE}$, then there exist $i \in \{1,\dots,s/2\}$ and $j \in \{s/2+1,\dots,s\}$ such that $x_{ij} = y_{ij} = 1$, and by the construction of $G_{x,y}$, both edges (a_i, a_j) and (b_i, b_j) are absent. Every path from a_i to b_j must go through some $(a' \rightsquigarrow b')$ path of length $k-1$, and if $\mathrm{dist}(a_i, b_j) \le k$ then the shortest path connecting a_i and b_j can only contain one more edge. However, since the edges (a_i, a_j) and (b_i, b_j) are both absent in $G_{x,y}$, no such path exists, so $\mathrm{dist}(a_i, b_j) > k$, which implies that $D(G_{x,y}) > k$.

Theorem 1. *For every k, the proof size of any PLS for $(\mathcal{F}, (D \le k))$ is $\Omega(n/k)$.*

Proof. Consider any PLS for $(\mathcal{F}, (D \le k))$, and construct a nondeterministic protocol for $\mathrm{DISJ}(x,y)$ as follows. Alice and Bob simulate the verification of $D(G_{x,y}) \le k$ using the PLS, such that Alice simulates the nodes in $A = A_1 \cup A_2 \cup a$, and Bob simulates the rest of the nodes, denoted by B. Each of the players nondeterministically chooses the labels of its nodes as his auxiliary bit-string. Alice and Bob then exchange the labels corresponding to the nodes touching the cut, and simulate the verification process in all nodes. Then, they compute a and b, the conjunction of the returned values of A and B respectively. Finally, Alice sends a to Bob, Bob sends b to Alice, and they both output the conjunction $a \wedge b$ as the solution for $\mathrm{DISJ}(x,y)$.

If $\mathrm{DISJ}(x,y) = \texttt{TRUE}$ then $D(G_{x,y}) \le k$, there is an assignment of labels to the nodes such that all nodes output $\texttt{TRUE}$, and if both players choose these labels as their bit-strings then they both output $\mathrm{DISJ}(x,y) = \texttt{TRUE}$. On the other hand, if $\mathrm{DISJ}(x,y) = \texttt{FALSE}$ then $D(G_{x,y}) > k$, for every assignment of labels to the nodes at least one node outputs $\texttt{FALSE}$, and Alice and Bob output $\mathrm{DISJ}(x,y) = \texttt{FALSE}$ in all executions.

Thus, the simulation we presented is a nondeterministic protocol for deciding $\mathrm{DISJ}(x,y)$. We know that in any nondeterministic protocol for Set-Disjointness $(s/2)^2$ elements, Alice and Bob must exchange $\Omega((s/2)^2)$ bits. The number of edges in the cut of $G_{x,y}$ induced by the partition of the nodes between Alice and Bob in the simulation is $s+1$. Therefore, the proof size of any PLS for $(\mathcal{F}, (D \le k))$ is $\Omega(s) \in \Omega(n/k)$.

We now show that in the APLS model there are schemes with much smaller proof size. We start with a 3/2-APLS and construct a scheme that is based on the randomized algorithm for a 3/2-approximation of the diameter presented in [30]. We use the following two lemmas.

Lemma 2. *Let $G = (V,E)$ be a graph, let $S, N \subseteq V$ be two sets of nodes, and consider a node $w \in V$. Assume that N is the set of z nodes closest to w for some parameter z, w is the farthest node from the set S, and $N \cap S$ is non-empty. Then, the largest depth D' of a BFS tree rooted at a node in $R = N \cup S \cup \{w\}$ satisfies $\frac{2}{3}D \le D' \le D$.*

Lemma 3. *Let $G = (V,E)$ be a graph and $z \in \mathbb{N}$ a parameter. For each $v \in V$, let $N_z(v)$ be the set of z nodes closest to v. Then, there exists a hitting set for $\{N_z(v) \mid v \in V\}$, of size $O(n \log n/z)$.*

Lemma 2 corresponds to an adapted version of Lemma 4 of [30], and Lemma 3 is a corollary of Theorem 2.7 of [2]. We obtain the following result.

Theorem 2. *There exists a 3/2-APLS for $(\mathcal{F}, (D \le k))$ with proof size $O(\sqrt{n}\log^2 n)$.*

Proof. Our scheme is based on Lemma 2: it consists of a node w, sets N and S and all the BFS trees rooted at $R = N \cup S \cup \{w\}$. In addition, there is a node w' that is used to verify that the largest depth of a BFS tree rooted in R is as claimed, and a BFS tree rooted at w'. The main task in our scheme is to verify the BFS trees described above, and to verify that the diameter estimation, i.e., the maximum depth of the trees, is at most k. Since a BFS tree verification is known from previous work, the challenges in the scheme construction is to verify locally that w is indeed the farthest node from the set S, that N is the neighborhood of w, and that the estimation is indeed the maximum depth of a tree.

Formally, let $G_s \in \mathcal{F}$ be a configuration with the underlying graph $G = (V,E)$ and $D(G_s) \le k$. For every $v \in V$, denote by $N_{\sqrt{n}}(v)$ the $\sqrt{n}$ nodes closest to v (break ties according to IDs), and let $S \subset V$ be a set of $O(\sqrt{n}\log n)$ nodes such that S hits $\left\{N_{\sqrt{n}}(v) \mid v \in V\right\}$, whose existence follows from Lemma 3.

Let $h(v) = \min\{\mathrm{dist}(v,u) \mid u \in S\}$, the distance of v from the set S, and let w be the farthest node from S, i.e., $h(w) \ge h(v)$ for every $v \in V$. Let $q(w)$ be the largest distance from w to any node in $N_{\sqrt{n}}(w)$. Let $R = S \cup \{w\} \cup N_{\sqrt{n}}(w)$ be a set of $|R| = O(\sqrt{n}\log n)$ nodes, and consider the set R_{BFS} of BFS trees rooted at nodes in R. Let $d_{\max}$ be the maximum depth of a tree in R_{BFS} and let w' be a node at distance $d_{\max}$ from one of the roots.

The label assigned to a node $v \in V$ is

$$\ell(v) = (\ell_{\mathrm{BFSs}-S}(v), \ell_{\mathrm{BFSs}-N}(v), \ell_{\mathrm{BFS}-w}(v), \ell_{\mathrm{BFS}-w'}(v), \ell_{hw}(v), \ell_{qw}(v), \ell_{\max-\mathrm{dist}}(v))$$

where $\ell_{\text{BFSs}-S}(v)$ is a set of $O(\sqrt{n}\log n)$ pairs $\{(\text{ID}(u), \text{dist}(v,u)) \mid u \in S\}$; $\ell_{\text{BFSs}-N}(v)$ is a set of $\sqrt{n}$ pairs $\big\{(\text{ID}(u), \text{dist}(v,u)) \mid u \in N_{\sqrt{n}}(w)\big\}$; $\ell_{\text{BFS}-w}(v) = (\text{ID}(w), \text{dist}(v,w))$; and $\ell_{\text{BFS}-w'}(v) = (\text{ID}(w'), \text{dist}(v,w'))$. Every pair mentioned above is the label needed in order to verify the correct structure of the corresponding BFS tree. In order to verify that w is indeed the farthest node from S, every node is given the distance of w from S, $\ell_{hw}(v) = h(w)$; To verify the consistency of $N_{\sqrt{n}}(w)$, every node is given the radius of this neighborhood $\ell_{qw}(v) = q(w)$; and $\ell_{\max-\text{dist}}(v) = d_{\max}$ is given in order to verify the existence and maximality of the estimation $d_{\max}$.

In the verification process, a node v exchanges labels with all its neighbors, and verifies the following conditions:

1. Consistency of global parameters: For every neighbor v' of v, it holds that $\ell_{hw}(v') = \ell_{hw}(v)$, $\ell_{qw}(v') = \ell_{qw}(v)$, and $\ell_{\max-\text{dist}}(v') = \ell_{\max-\text{dist}}(v)$
2. All distances are bounded by $d_{\max}$ and k: For every pair (ID, d) in $\ell_{\text{BFSs}-S}(v) \cup \ell_{\text{BFSs}-N}(v) \cup \{\ell_{\text{BFS}-w}(v)\} \cup \{\ell_{\text{BFS}-w'}(v)\}$, it holds that $0 \le d \le \ell_{\max-\text{dist}}(v) \le k$.
3. Existence of a BFS tree of depth $d_{\max}$: If $\ell_{\text{BFS}-w'}(v) = (\text{ID}(v), 0)$ then there exists a pair $(\text{ID}, d) \in \ell_{\text{BFSs}-S}(v) \cup \ell_{\text{BFSs}-N}(v) \cup \{\ell_{\text{BFS}-w}(v)\}$ such that $d = \ell_{\max-\text{dist}}(v)$.
4. Only one pair for each node in S and in $N_{\sqrt{n}}(w)$: For every two pairs $(\text{ID}, d), (\text{ID}', d') \in \ell_{\text{BFSs}-X}(v)$, for $X \in \{S, N\}$, if $d \neq d'$ then $\text{ID} \neq \text{ID}'$.
5. BFS structures: For every neighbor v' of v and $X \in \{S, N\}$, the following holds. There exists a pair $(\text{ID}, d) \in \ell_{\text{BFSs}-X}(v)$, for some d if and only if there exists a pair $(\text{ID}, d') \in \ell_{\text{BFSs}-X}(v')$ with the same ID and $d' \in \{d-1, d, d+1\}$. For $x \in \{w, w'\}$, $\ell_{\text{BFS}-x}(v) = (\text{ID}, d)$ for some d if and only if $\ell_{\text{BFS}-x}(v') = (\text{ID}, d')$ for $d' \in \{d-1, d, d+1\}$.
6. Existence of roots: For every $X \in \{S, N\}$ and pair $(\text{ID}, d) \in \ell_{\text{BFSs}-X}(v)$, if $d > 0$ then there exists a neighbor v' of v with $(\text{ID}, d-1) \in \ell_{\text{BFSs}-X}(v')$. For $x \in \{w, w'\}$, if $\ell_{\text{BFS}-x}(v) = (\text{ID}, d)$ and $d > 0$ then there exists a neighbor v' of v with $\ell_{\text{BFS}-x}(v') = (\text{ID}, d-1)$.
7. Unique roots: For every pair (ID, d) in $\ell_{\text{BFSs}-S}(v) \cup \ell_{\text{BFSs}-N}(v) \cup \{\ell_{\text{BFS}-w}(v)\} \cup \{\ell_{\text{BFS}-w'}(v)\}$, if $d = 0$ then $\text{ID} = \text{ID}(v)$.
8. Non-empty intersection of S and $N_{\sqrt{n}}(w)$: There exists a pair $(\text{ID}, d) \in \ell_{\text{BFSs}-S}(v) \cap \ell_{\text{BFSs}-N}(v)$.
9. Maximality and correctness of $h(w)$: There exists a pair $(\text{ID}, d) \in \ell_{\text{BFSs}-S}(v)$ such that $d \le \ell_{hw}(v)$, and if $\ell_{\text{BFS}-w}(v) = (\text{ID}(v), 0)$ then there exists no pair $(\text{ID}, d) \in \ell_{\text{BFSs}-S}(v)$ such that $d < \ell_{hw}(v)$.
10. The neighborhood of w: Let $\ell_{\text{BFS}-w}(v) = (\text{ID}, d)$. If $d < \ell_{qw}(v)$ then there exists a pair $(\text{ID}(v), 0) \in \ell_{\text{BFSs}-N}(v)$, and if $d > \ell_{qw}(v)$ then there exists no pair $(\text{ID}(v), 0) \in \ell_{\text{BFSs}-N}(v)$.

The completeness of this 3/2-APLS follows from the fact that if $D(G_s) \le k$ then the maximum depth of any BFS tree in G_s is at most k.

For the soundness, consider a configuration $G_s \in \mathcal{F}$ with the underlying graph $G = (V, E)$ and label assignment ℓ, and assume that all nodes output `TRUE`. By (1), all nodes have the same values ℓ_{hw}, ℓ_{qw} and $\ell_{\max-\text{dist}}$. By (4), (5), (6) and (7) for every node $v \in V$ and every pair $(\text{ID}, d) \in \ell_{\text{BFSs}-S}(v) \cup \ell_{\text{BFSs}-N}(v) \cup \{\ell_{\text{BFS}-w}(v)\} \cup \{\ell_{\text{BFS}-w'}(v)\}$, there exists a node u such that $\text{ID} = \text{ID}(u)$ and it holds that $d = \text{dist}(v, u)$.

Let $\overline{S}(v)$ be the collection of IDs in $\ell_{\text{BFSs}-S}(v)$, let $\overline{N}(v)$ be the collection of IDs in $\ell_{\text{BFSs}-N}(v)$, let $\overline{w}(v)$ be the ID in $\ell_{\text{BFS}-w}(v)$ and let $\overline{w'}(v)$ be the ID in $\ell_{\text{BFS}-w'}(v)$. By (5), for every two nodes v and u it holds that $\overline{S}(v) = \overline{S}(u)$, $\overline{N}(v) = \overline{N}(u)$, $\overline{w}(v) = \overline{w}(u)$ and $\overline{w'}(v) = \overline{w'}(u)$. We denote these values by $\overline{S}$, $\overline{N}$, $\overline{w}$ and $\overline{w'}$ respectively. By (10), $\overline{N}$ is the set of closest nodes to $\overline{w}$; by (9), $\overline{w}$ is the farthest node from the set $\overline{S}$; and by (8), there exists some node in the intersection of $\overline{N}$ and $\overline{S}$. By (3), the collection of pairs $\ell_{\text{BFS}-w'}(v)$ of all nodes $v \in V$ indicates a BFS rooted at $\overline{w_0}$ with distance $\ell_{\max-\text{dist}}$ to one of the nodes in $\overline{S} \cup \overline{N} \cup \{\overline{w}\}$, and by (2) we know that this is the largest distance from any node to one of the nodes in $\overline{S} \cup \overline{N} \cup \{\overline{w}\}$ and this distance is at most k.

Overall, we have a collection of BFS trees with depth at most $\ell_{\max-\text{dist}} \le k$. Therefore, all conditions of Lemma 2 are satisfied, and we have $(2/3)D(G_s) \le \ell_{\max-\text{dist}}$. Hence, $D(G_s) \le (3/2)k$ as desired.

The proof size of the scheme follows from Lemma 3, which implies that there exists a set S of size $O(\sqrt{n}\log n)$ that is a hitting set for $\big\{N_{\sqrt{n}}(v) \mid v \in V\big\}$. In particular, the intersection $N_{\sqrt{n}}(w) \cap S$, where w is the farthest node from S, is not empty. Therefore, the label consists of $O(\sqrt{n}\log n)$ sub-labels of size $O(\log n)$ each.

The following result shows that with the proof size we obtain for 3/2-APLS we cannot have a better approximation ratio that is correct for all possible bounds k. To get a better approximation ratio, one needs to use labels that are almost as large as the labels used for exact PLS.

Let x and y be two s-bit strings, $s \in \Omega(n/\log n)$. Our lower bound follows the recent construction of Abboud et al. [1].[5]

[5] See Chapter 2.2 of [1]. We use $P = \lfloor (k-2)/4 \rfloor$.

Lemma 4 ([1]). *Given two strings $x, y \in \{0,1\}^s$, there exists a graph $G_{x,y} = (V, E)$ and a partition of V into V_A and V_B such that:*

1. *The number of nodes in $G_{x,y}$ is $n \in \Theta(s \log s)$.*
2. *All the edges depending on x are between nodes in V_A.*
3. *All the edges depending on y are between nodes in V_B.*
4. *The number of edges between nodes in V_A and V_B is in $\Theta(\log s)$.*
5. *If $\mathrm{DISJ}(x, y) =$ TRUE then $D(G_{x,y}) \le k$, and otherwise $D(G_{x,y}) > 3k/2 - 9$.*

From this construction we derive the following lower bound.

Theorem 3. *For every k, there exists an $\epsilon \in \Theta(1/k)$ such that the proof size of any $(3/2 - \epsilon)$-APLS for $(\mathcal{F}, (D \le k))$ is $\Omega(n/\log^2 n)$.*

Proof. Consider a $(3/2 - 9/k)$-APLS for $(\mathcal{F}, (D \le k))$, and an instance (x, y) of the DISJ problem over s bits. Construct the graph $G_{x,y}$ as in Lemma 4, with the same partition to V_A and V_B. Alice and Bob nondeterministically choose the labels for the nodes of V_A and V_B, simulate the verification algorithm together, and then compute a and b, the conjunction of the returned values of V_A and V_B. Finally, Alice sends a to Bob, Bob sends b to Alice, and they both output the conjunction $a \wedge b$ as the solution for $\mathrm{DISJ}(x, y)$.

By Lemma 4, if $\mathrm{DISJ}(x, y) =$ TRUE then $D \le k$, all nodes must accept and Alice and Bob return TRUE. On the other hand, If $\mathrm{DISJ}(x, y) =$ FALSE then $D > (3/2 - 9/k)k$, at least one node rejects, and Alice and Bob return FALSE. Thus, Alice and Bob correctly solve the Set-Disjointness problem over s elements.

Note that $\log n = \Theta(\log s)$. Alice and Bob must communicate $\Omega(s) = \Omega(n/\log n)$ bits, and there are $O(\log n)$ nodes touching the cut, so the proof size is $\Omega(n/\log^2 n)$.

To further study the tradeoff between the approximation ratio and the proof size, we now prove that if we increase the approximation ratio we can construct an even more efficient scheme.

Theorem 4. *There exists a 2-APLS for $(\mathcal{F}, (D_w \le k))$ with proof size $O(\log n + \log W)$.*

Proof. Let $G_s \in \mathcal{F}$ such that $D_w(G_s) \le k$, and let $r \in V$ be some node. The label assigned to every node $v \in V$ is $\ell(v) = (\ell_{\mathrm{dist}}(v), \ell_{\mathrm{root}}(v))$, where $\ell_{\mathrm{dist}}(v) = \mathrm{dist}_w(r, v)$ and $\ell_{\mathrm{root}}(v) = \mathrm{ID}(r)$. To verify that $D_w(G_s) \le k$, a node v exchanges labels with all its neighbors, and verifies the following conditions:

1. For every neighbor u of v, it holds that $\ell_{\mathrm{root}}(u) = \ell_{\mathrm{root}}(v)$.
2. $0 \le \ell_{\mathrm{dist}}(v) \le k$.
3. If $\ell_{\mathrm{dist}}(v) > 0$ then v has at least one neighbor u with $\ell_{\mathrm{dist}}(u) = \ell_{\mathrm{dist}}(v) - w(u, v)$.
4. If $\ell_{\mathrm{dist}}(v) = 0$ then $\ell_{\mathrm{root}}(v) = \mathrm{ID}(v)$.

The completeness of this 2-APLS is clear: If $D_w(G_s) \le k$ and labels are assigned as described above, all nodes output TRUE.

For the soundness, consider a configuration G_s with label assignment ℓ, such that all nodes output TRUE. For a node v in the graph, follow the path from v constructed by repeatedly going from a node v' to its neighbor u with $\ell_{\mathrm{dist}}(u) = \ell_{\mathrm{dist}}(v') - w(u, v')$, whose existence is guaranteed by Condition (3). By conditions (2) and (3), this path must end after traversing a weight of at most k, at a node r with $\ell_{\mathrm{dist}}(r) = 0$, and this node is unique by conditions (1) and (4). As this claim can be applied to each node in the graph, every two nodes in the graph are connected to each other by a path through r, of weighted distance at most $2k$, and $D_w(G_s) \le 2k$ as desired.

The following corollary is a direct result of Theorem 4 for the unweighted case.

Corollary 1. *There exists a 2-APLS for $(\mathcal{F}, (D \le k))$ with proof size $O(\log n)$.*

4 Maximum Weight Matching

Given a configuration $G_s \in \mathcal{F}_M$ with the underlying graph $G = (V, E)$, an edge weight function w, and a specified matching $M \subset E$, we wish to verify $(\mathcal{F}_M, (w(M) \ge w(\mathrm{MWM})))$. Göös and Suomela [22] present a PLS for this problem in bipartite graphs, using a linear programming (LP) formulation. Here, we extend their technique to present a 2-APLS for $(\mathcal{F}_M, (w(M) \ge w(\mathrm{MWM})))$ on general graphs.

Our 2-APLS is simple: the label of a matched node is the weight of its matched edge, and the label of an unmatched node is 0. The verification process, and the proof that this is indeed a 2-APLS are slightly more involved, and use a relaxation of the complementary slackness conditions of a relaxation of a linear-programming formulation for the problem.

Consider the next integral-LP formulation of the MWM problem (cf. [10, Chapter 5]):

$$\begin{array}{ll} \text{Maximize} & \sum_{e\in E} w(e)x_e \\ \text{Subject to} & \sum_{\{e|v\in e\}} x_e \leq 1, \forall v \in V \\ & x_e \in \{0,1\}\,, \qquad \forall e \in E, \end{array}$$

and the LP obtained by relaxing the integrality condition into:

$$x_e \geq 0, \;\; \forall e \in E.$$

The dual linear-program of the relaxed problem is

$$\begin{array}{ll} \text{Minimize} & \sum_{v\in V} y_v \\ \text{Subject to} & y_u + y_v \geq w(e), \forall e = (u,v) \in E. \end{array}$$

Given a pair consisting of a primal and a dual feasible solutions, their optimality can be verified by checking several conditions derived from the LP, conditions that are known as the *complementary slackness conditions.* For the aforementioned LP, the conditions are:

$$\begin{array}{l} x_e > 0 \implies y_u + y_v = w(e), \;\; e = (u,v) \in E; \text{ and} \\ y_v > 0 \implies \sum_{\{e|v\in e\}} x_e = 1, \, v \in V. \end{array}$$

If G is bipartite, then any pair of feasible optimal solutions satisfy the complementary slackness conditions, a fact that lies at the heart of the PLS presented by Göös and Suomela [22].

For general graphs, the same method fails miserably. The inherent obstacle that this approach faces is the integrality gap of the LP formulation: a fractional solution to the problem may be twice as large as the maximum integral solution. While there are LP formulations of the problem with an integrality gap of 1, it is not clear how to translate them into a PLS, since the number of dual variables in these LPs is substantially larger.

However, we observe that a relaxed version of these conditions is enough to prove that a primal solution is an approximation of the MWM.

Theorem 5 (See [31, Section 15.1]). *If x and y are feasible primal and dual solutions in a graph G satisfying*

$$\begin{array}{ll} x_e > 0 \implies w(e) \leq y_u + y_v \leq 2w(e), \, e = (u,v) \in E; \textit{ and} \\ y_v > 0 \implies \sum_{\{e|v\in e\}} x_e = 1, \qquad\qquad v \in V, \end{array}$$

then x is a 2-approximation of the MWM in G.

Unlike the case of bipartite graphs, here the opposite implication does not hold: not every pair of 2-approximate solutions fulfill the conditions. Thus, given a matching represented by a vector x, we explicitly build a dual solution y such that x and y satisfy above conditions. This dual solution y will serve as a 2-APLS for $(\mathcal{F}_M, (w(M) \geq w(\text{MWM})))$ in a general graph.

Theorem 6. *There exists a 2-APLS for $(\mathcal{F}_M, (w(M) \geq w(\text{MWM})))$ with proof size $O(\log W)$.*

Proof. Let G be a weighted graph with weights in $\{1, \ldots, W\}$ and M a maximum weight matching in G. Let $(x_e)_{e\in E}$ be the indicator vector of M. Define the values of the dual variables $(y_v)_{v\in V}$ by $y_v = w(e)$ if there exist an edge $e \in M$ such that $v \in e$, and $y_v = 0$ otherwise. The label of a node v is set to be y_v.

To verify $(\mathcal{F}_M, (w(M) \geq w(\text{MWM})))$, a node v exchanges labels with its neighbors and check the next feasibility condition:

- For each neighbor u of v, $y_u + y_v \geq w(u,v)$.

We start by showing that if M is indeed a MWM, then the relaxed complementary slackness conditions hold. Let $e = (u,v)$ be an edge satisfying $x_e > 0$, i.e. $e \in M$, then $y_u = y_v = w(e)$ and indeed $w(e) \leq y_u + y_v \leq 2w(e)$. For the second complementary slackness condition, let v be a node with $y_v > 0$, so there is exactly one edge $(u,v) \in M$ with $x_{(u,v)} = 1$, while for every other neighbor u' of v, $x_{(u',v)} = 0$, so $\sum_{\{e|v\in e\}} x_e = 1$.

For the feasibility, the input is a feasible matching, so $\sum_{\{e|v\in e\}} x_e \leq 1$ for each node v and $x_e \geq 0$ for each edge e, and the primal solution x is feasible. For the dual solution y, assume towards contradiction that there is an edge $e = (u, v)$, $e \notin M$, such that $y_u + y_v < w(e)$. Then, the matching obtained by removing any edge in M that touches u or v and adding e to M has a weight $w(M) - (y_u + y_v) + w(e) > w(M)$, which contradicts the maximality of M. The case of $e \in M$ was considered in the previous paragraph. Thus, we have a pair of feasible primal and dual solutions satisfying the relaxed slackness conditions, and the solutions are 2-approximations of the optimal solutions.

Finally, consider a configuration G_s with label assignment (x_e), such that all nodes output `TRUE`. The labels represent a dual solution that satisfies all the relaxed complementary slackness conditions, so by Theorem 5 the solution is a 2-approximation of the MWM.

We are unaware of any lower bound for the MWM problem in the PLS model, nor in the CONGEST and LOCAL models. We note that for every approximation ratio $\alpha \geq 1$, some communication is needed in any α-APLS for $(\mathcal{F}_M, (w(M) \geq w(\text{MWM})))$. This is true since, for every configuration G_s with an empty matching $M = \emptyset$ (not any approximation of MWM), the local view of every node is consistent with some legal configuration with matching M', where $w(M') = w(\text{MWM})$. Let v be a node and let $u_1, \ldots, u_d$ be the neighbors of v where the weight of every edge (v, u_i) is w_i. The construction of the legal configuration G_s^v for v is as follows. Add nodes $z_1, \ldots, z_d$ and an edge $e_i = (z_i, u_i)$ of weight $w_i + 1$ for every $1 \leq i \leq d$. Finally, define $M' = \{e_i \mid 1 \leq i \leq d\}$. It is easy to verify that there is no augmenting path for M' in this configuration, i.e., $w(M') = w(\text{MWM})$. However, the local view of v in G_s and in G_s^v is the same. Therefore, without communication, v must output `TRUE`. Since the same holds for every node, we conclude that some communication is necessary, regardless of the desired approximation ratio.

5 Discussion

This paper presents the new model of approximate proof-labeling schemes. We illustrate the power of the APLS model with the $D \leq k$ predicate. We prove a tight lower bound (up to a logarithmic factor) in the PLS model, and present two, more efficient, APLSs for this predicate. The two APLSs show a non-trivial tradeoff between the approximation ratio and the proof size.

We also present a 2-APLS for the predicate $w(M) \geq w(\text{MWM})$ on general graphs, a problem for which it is unknown if a non-trivial PLS exists. Presenting an efficient PLS for this problem, showing that a PLS with small proof size does not exist, or presenting an APLS with different approximation ratio or different proof size are interesting questions left open.

It would be interesting to study the APLS model on other graph predicates. For example, the chromatic number $\chi(G)$ of a graph G is the minimal number of colors in a proper node coloring of G. A PLS for $\chi \leq k$ with proof size $O(\log k)$ exists, where the proof is a proper coloring of the graph. However, it was proven in [22] that any PLS for $\chi > 3$ must have $\tilde{\Omega}(n^2)$ proof size. Hence, also for this problem, the APLS model may allow a more efficient verification.

Finally, the idea of approximation in verification we present in this paper can be extended to other decision and verification schemes, such as the complexity classes LD and NLD, generating a different classification of problems. For example, our 2-APLS for $w(M) \geq w(\text{MWM})$ on general graphs can also be used for 2-approximate NLD, under the relevant definitions, since the labels can be locally computed by the nodes.

Acknowledgment We thank Gilad Kutiel, Seffi Naor and Dror Rawitz for discussions about the primal-dual method, and the anonymous reviewers of SIROCCO 2017 for valuable comments.

References

1. A. Abboud, K. Censor-Hillel, and S. Khoury. Near-linear lower bounds for distributed distance computations, even in sparse networks. In *Distributed Computing - 30th International Symposium, DISC*, pages 29–42, 2016.
2. D. Aingworth, C. Chekuri, P. Indyk, and R. Motwani. Fast estimation of diameter and shortest paths (without matrix multiplication). *SIAM J. Comput.*, 28(4):1167–1181, 1999.
3. H. Arfaoui, P. Fraigniaud, D. Ilcinkas, and F. Mathieu. Distributedly testing cycle-freeness. In *40th Int. Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, LNCS, pages 15–28. Springer, 2014.
4. H. Arfaoui, P. Fraigniaud, and A. Pelc. Local decision and verification with bounded-size outputs. In *15th Symp. on Stabilization, Safety, and Security of Distributed Systems (SSS)*, LNCS, pages 133–147. Springer, 2013.
5. B. Awerbuch, B. Patt-Shamir, and G. Varghese. Self-stabilization by local checking and correction. In *32nd Symposium on Foundations of Computer Science (FOCS)*, pages 268–277. IEEE, 1991.

6. M. Baruch, P. Fraigniaud, and B. Patt-Shamir. Randomized proof-labeling schemes. In *Proc. 34th ACM Symp. on Principles of Distributed Computing (PODC)*, pages 315–324, 2015.
7. M. Baruch, R. Ostrovsky, and W. Rosenbaum. Space-time tradeoffs for distributed verification. *CoRR*, abs/1605.06814, 2016.
8. L. Blin, P. Fraigniaud, and B. Patt-Shamir. On proof-labeling schemes versus silent self-stabilizing algorithms. In *16th Int. Symp. on Stabilization, Safety, and Security of Distributed Systems (SSS)*, LNCS, pages 18–32. Springer, 2014.
9. S. Chechik, D. H. Larkin, L. Roditty, G. Schoenebeck, R. E. Tarjan, and V. V. Williams. Better approximation algorithms for the graph diameter. In *ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 1041–1052, 2014.
10. W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver. *Combinatorial Optimization.* John Wiley & Sons, Inc., New York, NY, USA, 1998.
11. A. Das Sarma, S. Holzer, L. Kor, A. Korman, D. Nanongkai, G. Pandurangan, D. Peleg, and R. Wattenhofer. Distributed verification and hardness of distributed approximation. *SIAM J. Comput.*, 41(5):1235–1265, 2012.
12. L. Feuilloley and P. Fraigniaud. Survey of distributed decision. *Bulletin of the EATCS*, 119, 2016.
13. L. Feuilloley, P. Fraigniaud, and J. Hirvonen. A Hierarchy of Local Decision. In *43rd Int. Colloquium on Automata, Languages, and Programming, ICALP*, volume 55 of *LIPIcs*, pages 118:1–118:15. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016.
14. K.-T. Foerster, T. Luedi, J. Seidel, and R. Wattenhofer. Local checkability, no strings attached. In *International Conference on Distributed Computing and Networking, ICDCN*, pages 21:1–21:10. ACM, 2016.
15. K.-T. Foerster, O. Richter, J. Seidel, and R. Wattenhofer. Local checkability in dynamic networks. In *International Conference on Distributed Computing and Networking, ICDCN*, pages 4:1–4:10. ACM, 2017.
16. P. Fraigniaud, M. Göös, A. Korman, and J. Suomela. What can be decided locally without identifiers? In *ACM Symposium on Principles of Distributed Computing, PODC*, pages 157–165. ACM, 2013.
17. P. Fraigniaud, M. M. Halldórsson, and A. Korman. On the impact of identifiers on local decision. In *Principles of Distributed Systems: 16th International Conference, OPODIS*, pages 224–238. Springer Berlin Heidelberg, 2012.
18. P. Fraigniaud, J. Hirvonen, and J. Suomela. Node labels in local decision. In *Structural Information and Communication Complexity: 22nd International Colloquium, SIROCCO*, pages 31–45. Springer, 2015.
19. P. Fraigniaud, A. Korman, and D. Peleg. Towards a complexity theory for local distributed computing. *J. ACM*, 60(5):35, 2013.
20. P. Fraigniaud, S. Rajsbaum, and C. Travers. Locality and checkability in wait-free computing. *Distributed Computing*, 26(4):223–242, 2013.
21. P. Fraigniaud, S. Rajsbaum, and C. Travers. On the number of opinions needed for fault-tolerant run-time monitoring in distributed systems. In *5th Int. Conf. on Runtime Verification*, LNCS, pages 92–107. Springer, 2014.
22. M. Göös and J. Suomela. Locally checkable proofs in distributed computing. *Theory of Computing*, 12(1):1–33, 2016.
23. S. Holzer, D. Peleg, L. Roditty, and R. Wattenhofer. Distributed 3/2-approximation of the diameter. In *Distributed Computing - 28th International Symposium, DISC*, pages 562–564, 2014.
24. S. Holzer and R. Wattenhofer. Optimal distributed all pairs shortest paths and applications. In *ACM Symposium on Principles of Distributed Computing, PODC*, pages 355–364, 2012.
25. A. Korman and S. Kutten. Distributed verification of minimum spanning trees. *Distributed Computing*, 20:253–266, 2007.
26. A. Korman, S. Kutten, and T. Masuzawa. Fast and compact self stabilizing verification, computation, and fault detection of an MST. In *ACM Symposium on Principles of Distributed Computing, PODC*, pages 311–320, 2011.
27. A. Korman, S. Kutten, and D. Peleg. Proof labeling schemes. *Distributed Computing*, 22(4):215–233, 2010.
28. E. Kushilevitz and N. Nisan. *Communication Complexity.* Cambridge University Press, New York, NY, USA, 1997.
29. D. Peleg, L. Roditty, and E. Tal. Distributed algorithms for network diameter and girth. In *Automata, Languages, and Programming - 39th International Colloquium, ICALP*, pages 660–672, 2012.
30. L. Roditty and V. V. Williams. Fast approximation algorithms for the diameter and radius of sparse graphs. In *Symposium on Theory of Computing Conference, STOC*, pages 515–524, 2013.
31. V. V. Vazirani. *Approximation algorithms.* Springer, 2001.