Approximation Algorithms and Hardness Results for Labeled Connectivity Problems

Refael Hassin[1], Jérôme Monnot[2], and Danny Segev[1]

[1] School of Mathematical Sciences, Tel-Aviv University, Israel.
{hassin,segevd}@post.tau.ac.il
[2] CNRS LAMSADE, Université Paris-Dauphine, France.
monnot@lamsade.dauphine.fr

Abstract. Let $G = (V, E)$ be a connected multigraph, whose edges are associated with labels specified by an integer-valued function $\mathcal{L} : E \to \mathbb{N}$. In addition, each label $\ell \in \mathbb{N}$ to which at least one edge is mapped has a non-negative cost $c(\ell)$. The *minimum label spanning tree* problem (MinLST) asks to find a spanning tree in G that minimizes the overall cost of the labels used by its edges. Equivalently, we aim at finding a minimum cost subset of labels $I \subseteq \mathbb{N}$ such that the edge set $\{e \in E : \mathcal{L}(e) \in I\}$ forms a connected subgraph spanning all vertices. Similarly, in the *minimum label s-t path* problem (MinLP) the goal is to identify an s-t path minimizing the combined cost of its labels, where s and t are provided as part of the input.
The main contributions of this paper are improved approximation algorithms and hardness results for MinLST and MinLP. As a secondary objective, we make a concentrated effort to relate the algorithmic methods utilized in approximating these problems to a number of well-known techniques, originally studied in the context of integer covering.

1 Introduction

The majority of graph connectivity problems have traditionally been studied under the assumption that each edge is associated with a *numerical attribute*, to which we refer as length, weight or cost, depending on the related real-life context. In this long-established model, the computational task is to identify a subgraph satisfying given connectivity requirements, with the objective of minimizing some function defined over the lengths of picked edges. While these settings capture a wide range of practical scenarios, they nevertheless fail to incorporate *grouping constraints* stating that the set of available edges is partitioned into classes, each of which can be purchased in its entirety or not at all. A rather convenient way of integrating grouping constraints is to couple each edge with a *label* that specifies its class. Having this extra notation at hand, we say that a subset of labels forms a feasible solution when the edges whose labels belong to this subset induce a subgraph satisfying the given connectivity requirements. Since costs are now assigned to labels rather than to single edges, the objective is to find a solution that minimizes some function defined over the costs of picked labels.

We address two of the most fundamental labeled connectivity problems, those of constructing spanning trees and s-t paths by picking labels of minimum total cost. Formally, let $G = (V, E)$ be a connected multigraph on n vertices, whose edges are associated with labels specified by an integer-valued function $\mathcal{L} : E \to \mathbb{N}$. In addition, each label $\ell \in \mathbb{N}$ to which at least one edge is mapped has a non-negative cost $c(\ell)$. The *minimum label spanning tree* problem (MinLST) asks to find a spanning tree in G that minimizes the overall cost of the labels used by its edges. Equivalently, we aim at finding a minimum cost subset of labels $I \subseteq \mathbb{N}$ such that the edge set $\{e \in E : \mathcal{L}(e) \in I\}$ forms a connected subgraph spanning all vertices. Similarly, in the *minimum label s-t path* problem (MinLP) the goal is to identify an s-t path minimizing the combined cost of its labels, where s and t are provided as part of the input. We refer to the special cases of these problems in which at most r edges are assigned to any given label as MinLST_r and MinLP_r, respectively.

1.1 Related results

Prior to describing the line of work preceding the current paper, we remark that, to the best of our knowledge, the weighted version of both MinLST and MinLP has not been previously studied. Therefore, the reader should bear in mind that the undermentioned upper and lower bounds on the approximability of these problems are stated with respect to the unweighted case, in which each label has a unit cost.

Chang and Leu [9] seem to have been the first to consider MinLST. They proved that the corresponding decision problem is NP-complete, and experimentally studied the performance of several heuristics, one of which is the *maximum vertex covering* algorithm. Krumke and Wirth [16] demonstrated that a variant of this algorithm (henceforth, modified MVC) guarantees an approximation factor of at most $2 \ln n + 1$, and accompanied this result by a hardness proof showing that MinLST is at least as hard to approximate as *set cover*. Wan, Chen and Xu [21] suggested a refined analysis of the modified MVC algorithm to obtain a factor of at most $\mathcal{H}_{n-1}$. Very recently, Xiong, Golden and Wasil [23] established that this algorithm provides a tight approximation guarantee of $\mathcal{H}_r$ for MinLST_r, improving the bound of Wan et al.[3], which is independent of r. Brüggemann, Monnot and Woeginger [7] considered a local-search heuristic, and showed that it constructs a solution for MinLST_r whose cost is within factor $\frac{r+1}{2}$ of optimum. In addition, they proved that MinLST_2 is polynomial-time solvable, whereas MinLST_r is APX-complete for $r \geq 3$.

Carr, Doddi, Konjevod and Marathe [8] proved that MinLP contains as a special case the *red-blue set cover* problem, which was shown in the same paper to be inapproximable within a factor of $O(2^{\log^{1-\epsilon} n})$ for any $\epsilon > 0$, unless $\text{NP} \subseteq \text{TIME}(n^{\text{polylog}(n)})$. However, this hardness result does not readily extend to MinLP, since the reduction described by Carr et al. is not approximation preserving. Relying on a more restrictive subproblem of red-blue set cover, Wirth [22, Thm. 2.16] established the abovementioned lower bound for MinLP. On the positive side, Broersma, Li, Woeginger and Zhang [6] devised two exact exponential-time algorithms, with respective running times of $O(n \cdot \min\{L^d, 2^L\})$ and $O(n^2 L!)$, where L is the number of labels and d is the s-t distance in G. They also considered a Dijkstra-like algorithm for approximating MinLP,

[3] Note that we may assume that $r \leq n - 1$, since in the opposite case MinLST_r can be reduced to MinLST_{n-1} by eliminating an edge from each uniform labeled cycle.

and demonstrated that it does not provide any constant factor. In fact, simple examples show that the resulting solution may have a cost of $\Omega(n)$ times the optimum, and moreover, to our knowledge a non-trivial approximation for MinLP has not been presented yet.

1.2 Our results

In this paper, we present improved approximation algorithms and hardness results for MinLST and MinLP. As a secondary objective, we make a concentrated effort to relate the algorithmic methods utilized in approximating these problems to a number of well-known techniques, originally studied in the context of integer covering. Our main findings can be briefly summarized as follows:

1. We extend the modified MVC algorithm to handle label costs in MinLST. Consequently, we derive the first algorithm for the weighted case, and prove that its approximation guarantee is $\mathcal{H}_{n-1}$. This result appears in Section 2.
2. We provide an additional $O(\log n)$ approximation for MinLST, which is based on assembling partial solutions obtained by repeatedly calling a constant-factor *maximum coverage* subroutine [1, 15, 20]. This approach encapsulates the principal idea we employ to approximate MinLP, and its specifics are described in Section 2.
3. By prematurely terminating the modified MVC heuristic and switching to an exact algorithm for MinLST_2 (due to Brüggemann et al. [7]), we achieve an approximation factor of $\mathcal{H}_r - \frac{1}{6}$ for unweighted MinLST_r. Our algorithm was inspired by a similar improvement for the set cover problem, proposed by Goldschmidt, Hochbaum and Yu [11]. In addition to showing that the factor $\mathcal{H}_r$ can be decreased by lower order terms, the underlying analysis we present is considerably simpler than that of Xiong et al. [23]. This algorithm is given in the full version of this paper [13].
4. We devise the first non-trivial algorithm for MinLP, with an approximation factor of $O(\sqrt{n})$. A crucial ingredient of this algorithm is a preprocessing step, in which we "guess" certain attributes of an arbitrary optimal solution and modify the given instance accordingly. Once again, we make use of repeated calls to a maximum coverage subroutine, eventually allowing us to easily identify a near-optimal solution. This result is described in Section 3.
5. Since MinLP_r admits a constant-factor approximation when $r = O(1)$, one may ask whether MinLP_r can be approximated in this case to any required degree. A negative answer to this question is provided in Section 4. Specifically, we show that MinLP_r is at least as hard to approximate as Min-r-SAT, a special case of the *minimum satisfiability* problem (MinSAT) in which each clause consists of at most r literals. The inapproximability of the former problem was studied by Avidor and Zwick [4], whereas that of MinSAT was studied even earlier by Marathe and Ravi [18].
6. By utilizing a self-improvability property of MinLP, which is based on the notion of *label squaring*, we show that MinLP cannot be approximated to within any polylogarithmic factor unless P = NP. This result is incomparable with the previously mentioned lower bound of Wirth, stating that an approximation factor of $O(2^{\log^{1-\epsilon} n})$

for some $\epsilon > 0$ implies NP $\subseteq$ TIME($n^{\text{polylog}(n)}$). Our technique was motivated by an analogous construction due to Karger, Motwani and Ramkumar [14] for the longest path problem. This proof is given in Section 4.

Due to space limitations, most proofs are omitted from this extended abstract. We refer the reader to the full version of this paper [13], in which all missing proofs are provided.

1.3 Notation

We conclude this section by introducing some notation and terminology. Given a set of edges $F \subseteq E$, we use $\mathcal{L}(F) = \{\mathcal{L}(e) : e \in F\}$ to denote the image of F under $\mathcal{L}$. Furthermore, when H is a subgraph of G, the notation $\mathcal{L}(H)$ is used as a shorthand for $\mathcal{L}(E(H))$. For a subset $I \subseteq \mathbb{N}$, we denote by $\mathcal{L}^{-1}(I) = \{e \in E : \mathcal{L}(e) \in I\}$ the inverse image of I, excluding the case where the specified subset is actually a singleton, which is abbreviated by writing $\mathcal{L}^{-1}(\ell)$ instead of $\mathcal{L}^{-1}(\{\ell\})$. The *contraction* of an edge (u, v) is the multigraph obtained by identifying the vertices u and v, followed by eliminating any degenerate edge joining the newly created vertex to itself. It is easy to verify that, regardless of the order according to which the edges in a subset $F \subseteq E$ are contracted, we always attain the same multigraph. Therefore, it is sensible to define the contraction of an edge set.

2 Approximating the Weighted MinLST Problem

In what follows, we present two approximation algorithms for the MinLST problem in its utmost generality, where each label has a non-negative cost. Guided by considerably different techniques, both algorithms iteratively construct a feasible subset of labels whose cost is within factor $O(\log n)$ of optimum. We remind the reader that MinLST is at least as hard to approximate as set cover, implying that the factor we derive is best possible up to a constant multiplicative factor, assuming P $\neq$ NP [3, 19].

2.1 The greedy algorithm

We extend the modified MVC algorithm, originally suggested by Krumke and Wirth [16], to handle label costs. In each step, our algorithm picks the most cost-effective label, namely, one that minimizes the ratio between its cost and the decrement in the number of vertices resulting from the contraction of its corresponding edges. A formal description of the algorithm is provided in Figure 1, followed by a tight analysis showing that its approximation guarantee is exactly $\mathcal{H}_{n-1}$.

Theorem 1. *The cost of the constructed solution is within factor $\mathcal{H}_{n-1}$ of optimum.*

Proof. Let $\{\ell_1, \dots, \ell_k\}$ be the set of labels returned by the algorithm, indexed by the order in which they were picked. In addition, for $1 \leq j \leq k$, let H_j be the processed multigraph at the beginning of the jth iteration (in which the label ℓ_j was picked). In what follows, we denote by OPT the cost of an optimal solution to the original

1. $I \leftarrow \emptyset$, $H \leftarrow G$.
2. While H contains at least two vertices
   (a) For every label $\ell \in \mathcal{L}(H)$, let $d_H(\ell)$ be the decrement in the number of vertices in H when the edge set $\mathcal{L}^{-1}(\ell)$ is contracted.
   (b) Pick a label ℓ^* that minimizes the ratio $\frac{c(\ell)}{d_H(\ell)}$ over all labels in $\mathcal{L}(H)$.
   (c) $I \leftarrow I \cup \{\ell^*\}$, $H \leftarrow$ the contraction of $\mathcal{L}^{-1}(\ell^*)$ in H.
3. Return I.

Fig. 1. The greedy algorithm

instance, and by $\mathrm{OPT}(H_j)$ the cost of an optimal solution to the instance we obtain at the beginning of the jth iteration. Clearly, $\mathrm{OPT} = \mathrm{OPT}(H_1) \geq \cdots \geq \mathrm{OPT}(H_k)$.

We first show that $c(\ell_j) \leq d_{H_j}(\ell_j)\frac{\mathrm{OPT}(H_j)}{|V(H_j)|-1}$ for all $1 \leq j \leq k$. Let $\{\ell_1^*, \ldots, \ell_p^*\} \subseteq \mathcal{L}(H_j)$ be an optimal solution to the instance corresponding to $\mathrm{OPT}(H_j)$. Note that the algorithm had the option of picking each ℓ_i^* when ℓ_j was picked. By observing that a minimum-ratio label is picked in each iteration, we have $\frac{c(\ell_j)}{d_{H_j}(\ell_j)} \leq \frac{c(\ell_i^*)}{d_{H_j}(\ell_i^*)}$ for every $1 \leq i \leq p$, and the stated upper bound on $c(\ell_j)$ follows as

$$\mathrm{OPT}(H_j) = \sum_{i=1}^{p} c(\ell_i^*) \geq \frac{c(\ell_j)}{d_{H_j}(\ell_j)} \sum_{i=1}^{p} d_{H_j}(\ell_i^*) \geq \frac{c(\ell_j)}{d_{H_j}(\ell_j)} \Big(|V(H_j)| - 1\Big) \ .$$

The second inequality holds since the set of edges $\mathcal{L}^{-1}(\{\ell_1^*, \ldots, \ell_p^*\})$ forms a connected subgraph spanning $V(H_j)$, implying that $\sum_{i=1}^{p} d_{H_j}(\ell_i^*) \geq |V(H_j)| - 1$.

Using the upper bounds proved above, we conclude that

$$\sum_{j=1}^{k} c(\ell_j) \leq \sum_{j=1}^{k} d_{H_j}(\ell_j) \frac{\mathrm{OPT}(H_j)}{|V(H_j)| - 1} \leq \sum_{j=1}^{k} \sum_{i=1}^{d_{H_j}(\ell_j)} \frac{\mathrm{OPT}}{|V(H_j)| - i} = \mathcal{H}_{n-1} \cdot \mathrm{OPT} \ ,$$

where the last equality holds since $d_{H_j}(\ell_j) = |V(H_j)| - |V(H_{j+1})|$. □

Lemma 2. *There are MinLST instances for which the algorithm produces a solution whose cost is $\mathcal{H}_{n-1}$ times the optimum.*

2.2 The budgeted covering algorithm

Unlike the shortsighted approach employed by the greedy algorithm, that picks a single label in each step, the new strategy we suggest consists of repeatedly contracting an inexpensive collection of labels in an attempt to decrease the number of vertices by a constant fraction. Such a collection is identified by approximating a related instance of the budgeted maximum coverage problem, in which we are given a ground set U, a family $\mathcal{S}$ of subsets of U with non-negative costs, and a budget B. The objective is to find a subcollection $\mathcal{S}' \subseteq \mathcal{S}$ such that the total cost of the subsets in $\mathcal{S}'$ is at most B, and such that the number of elements covered by $\mathcal{S}'$ is maximized. Several algorithms achieve an approximation guarantee of $1 - \frac{1}{e}$ for the latter problem [1, 15, 20].

To simplify the description and analysis of the budgeted covering algorithm, given in Figure 2, it would be convenient to make two preliminary assumptions. First, we assume that $c_{\min} = \min_{\ell \in \mathcal{L}(G)} c(\ell) > 0$, as all zero cost labels can be picked and contracted in advance. Second, given an accuracy requirement $\epsilon > 0$, we assume that a parameter $\Delta \in [\mathrm{OPT}, (1+\epsilon)\mathrm{OPT}]$ is known. This follows from observing that $c_{\min} \leq \mathrm{OPT} \leq |\mathcal{L}(G)| c_{\max}$, where $c_{\max} = \max_{\ell \in \mathcal{L}(G)} c(\ell)$, so all $O(\log_{1+\epsilon} \frac{|\mathcal{L}(G)| c_{\max}}{c_{\min}})$ candidate values of the form $(1+\epsilon)^k c_{\min}$ can be tested as the correct guess for Δ.

1. $I \leftarrow \emptyset$, $H \leftarrow G$.
2. While H contains at least two vertices
   (a) Create a budgeted maximum coverage instance by: The ground set is $V(H)$; for each label $\ell \in \mathcal{L}(H)$ there is a corresponding subset $V_\ell \subseteq V(H)$, consisting of all endpoints of edges in $\mathcal{L}^{-1}(\ell)$; the cost of V_ℓ is $c(\ell)$; and the budget is Δ.
   (b) Approximate the instance defined above, to obtain a subset $I' \subseteq \mathcal{L}(H)$.
   (c) $I \leftarrow I \cup I'$, $H \leftarrow$ the contraction of $\mathcal{L}^{-1}(I')$ in H.
3. Return I.

Fig. 2. The budgeted covering algorithm

Theorem 3. *The cost of the solution constructed by the budgeted covering algorithm is within factor* $(1+\epsilon)\log_{10/7} n$ *of optimum.*

Proof. Starting with an empty set of labels, in each iteration we augment I with labels whose total cost is at most $\Delta \leq (1+\epsilon)\mathrm{OPT}$. Therefore, it is sufficient to show that the algorithm terminates within $\log_{10/7} n$ iterations. To this end, we argue that contracting each of the label sets we obtain in step 2b decreases the number of vertices in the processed multigraph by a factor of at least 0.3.

Let $I^* \subseteq \mathcal{L}(G)$ be an optimal solution, with $\sum_{\ell \in I^*} c(\ell) = \mathrm{OPT} \leq \Delta$. Now consider a single iteration. Since $\mathcal{L}^{-1}(I^*)$ forms a connected subgraph of G spanning all vertices, it follows that $\{V_\ell : \ell \in I^* \cap \mathcal{L}(H)\}$ is a feasible solution to the budgeted maximum coverage instance defined in step 2a that fully covers $V(H)$. Consequently, for the current approximate solution I' we must have $|\bigcup_{\ell \in I'} V_\ell| \geq (1 - \frac{1}{e})|V(H)|$, implying that the contraction of $\mathcal{L}^{-1}(I')$ decreases the number of vertices by at least $\frac{1}{2}(1 - \frac{1}{e})|V(H)| > 0.3|V(H)|$. □

3 An $O(\sqrt{n})$ Approximation for MinLP

In what follows, we present the first non-trivial algorithm for the MinLP problem, achieving an approximation factor of $O(\sqrt{n})$. Throughout this section, we assume that the reader is familiar with the basics of budgeted maximum coverage given in Subsection 2.2.

The principal idea that guides our algorithm can be informally described as follows. When s and t are distant enough, an optimal solution must traverse many vertices, a fact that establishes the existence of an inexpensive set of labels whose contraction significantly decreases the number of vertices. As demonstrated in the context of the budgeted covering algorithm, we can identify a label set possessing this property by employing a maximum coverage subroutine. In the opposite case, a shortest path connecting s and t constitutes a near-optimal solution, provided that its edges are not endowed with overly priced labels. These observations suggest a two-step approach: First, perform repeated contractions as long as s and t are distant, and then complete the solution by picking a shortest s-t path.

For this tactic to have a low order strongly-polynomial running time, we apply a technique that was originally proposed by Hassin [12] and enhanced by Lorenz and Raz [17]. In adherence to standard terminology, we define an *α-test* to be a procedure that, given a parameter $\Delta \geq 0$, either constructs a feasible solution whose cost is at most $\alpha\Delta$ or determines that OPT $> \Delta$. The specifics of a $\frac{13}{3}\sqrt{n}$-test are provided in Figure 3, followed by a correctness proof.

1. $I \leftarrow \emptyset$, $H \leftarrow G$.
2. Eliminate from H all edges e with $c(\mathcal{L}(e)) > \Delta$.
3. While $\mathrm{dist}_H(s,t) \geq \sqrt{n}$
   (a) Create a budgeted maximum coverage instance by: The ground set is $V(H)$; for each label $\ell \in \mathcal{L}(H)$ there is a corresponding subset $V_\ell \subseteq V(H)$, consisting of all endpoints of edges in $\mathcal{L}^{-1}(\ell)$; the cost of V_ℓ is $c(\ell)$; and the budget is Δ.
   (b) Approximate the instance defined above, to obtain a subset $I' \subseteq \mathcal{L}(H)$.
   (c) $I \leftarrow I \cup I'$, $H \leftarrow$ the contraction of $\mathcal{L}^{-1}(I')$ in H.
4. If the number of iterations in step 3 was greater than $\frac{10}{3}\sqrt{n}$, return "OPT $> \Delta$".
5. Let P be a shortest s-t path in H. Return $I \cup \mathcal{L}(P)$.

Fig. 3. The MinLP test

Lemma 4. *The above procedure is a $\frac{13}{3}\sqrt{n}$-test.*

Now let c_{st} be the minimum label cost for which the subgraph $(V, \{e : c(\mathcal{L}(e)) \leq c_{st}\})$ contains an s-t path. Clearly, $c_{st} \leq \mathrm{OPT} \leq |\mathcal{L}(G)|c_{st}$. Given an accuracy requirement $\epsilon > 0$, we conduct a binary search over $\{(1+\epsilon)^k c_{st} : 0 \leq k \leq \lceil \log_{1+\epsilon} |\mathcal{L}(G)| \rceil\}$, involving $O(\log \log_{1+\epsilon} |\mathcal{L}(G)|)$ calls to the $\frac{13}{3}\sqrt{n}$-test described above. As a consequence, we identify a value Δ for which the test reports OPT $> \Delta$, whereas for $(1+\epsilon)\Delta$ it successfully constructs a feasible solution. It follows that the cost of this solution is at most $(1+\epsilon)\frac{13}{3}\sqrt{n} \cdot \mathrm{OPT}$.

Theorem 5. *For any fixed $\epsilon > 0$, MinLP can be approximated to within a factor of $(1+\epsilon)\frac{13}{3}\sqrt{n}$.*

4 The Hardness of Approximating MinLP

The main result of this section is a hardness proof showing that MinLP cannot be approximated to within any polylogarithmic factor unless P = NP. Prior to presenting this proof, we relate the approximability of MinLP_r with that of the Min-r-SAT problem.

4.1 MinLP_r and Min-r-SAT

The input to the *minimum satisfiability* problem (MinSAT) is a Boolean formula in conjugative normal form, consisting of a collection $C = \{C_1, \dots, C_m\}$ of clauses made up of complemented and uncomplemented occurrences of variables from the set $X = \{x_1, \dots, x_n\}$. The objective is to find a truth assignment to the variables that minimizes the number of satisfied clauses. We refer to the special case of this problem, in which each clause has at most r literals, as Min-r-SAT.

Marathe and Ravi [18] showed that MinSAT and vertex cover are equivalent with respect to approximability. Therefore, it is NP-hard to approximate the general MinSAT problem to within any factor smaller than 1.3606 [10]. Having observed that this bound does not apply to Min-r-SAT for small values of r, Avidor and Zwick [4] provided a lower bound of $\frac{15}{14}$ for $r = 2$, and a bound of $\frac{7}{6}$ for all $r \geq 3$. The next theorem extends these results to MinLP_r.

Theorem 6. *For every $r \geq 2$, $MinLP_r$ is at least as hard to approximate as Min-r-SAT.*

Proof. Given an instance (C, X) of Min-r-SAT, we show how to formulate it as a MinLP_r instance. For $1 \leq j \leq n$, let d_j and $\bar{d}_j$ be the number of clauses in which the literals x_j and $\bar{x}_j$ appear, respectively. Without loss of generality, $d_j \geq 1$ and $\bar{d}_j \geq 1$, or otherwise the value of x_j can be determined in advance. We define a MinLP_r instance $(G, \mathcal{L}, s, t)$ as follows:

1. The vertices of G are $v_1, \dots, v_{n+1}$. In addition, for every $1 \leq j \leq n$, we create two interior-disjoint paths, P_j and $\bar{P}_j$, connecting v_j and v_{j+1}. The length of P_j is d_j, while that of $\bar{P}_j$ is $\bar{d}_j$.
2. We now spread the labels $\{\ell_1, \dots, \ell_m\}$ on the edges of G. Specifically, let $C(x_j)$ and $C(\bar{x}_j)$ be the sets of clauses in C containing the literals x_j and $\bar{x}_j$, respectively. Then, each edge of P_j is given a distinct label from $\{\ell_i : C_i \in C(x_j)\}$, and similarly, the edges of $\bar{P}_j$ are given distinct labels from $\{\ell_i : C_i \in C(\bar{x}_j)\}$. Since each clause has at most r literals, the number of occurrences of each label is at most r.
3. Finally, we set $s = v_1$ and $t = v_{n+1}$.

We note that there is a one-to-one correspondence between truth assignments and s-t paths in G. First, suppose that f is a truth assignment that satisfies k clauses. Then the concatenation P of the paths $\{P_j : f(x_j) = \mathit{true}\}$ and $\{\bar{P}_j : f(x_j) = \mathit{false}\}$ forms an s-t path with $|\mathcal{L}(P)| = k$. Conversely, suppose that P is an s-t path with $|\mathcal{L}(P)| = k$. Then, as a result of setting each variable x_j to *true* if and only if P_j is a subpath of P, we obtain an assignment satisfying k clauses. □

4.2 Inapproximability within any polylogarithmic factor

In what follows, we prove that it is NP-hard to approximate MinLP within a factor of $O(\log^k n)$, for any fixed $k \geq 1$. To simplify the presentation, our proof is decomposed into three stages. First, we provide a logarithmic lower bound on the approximability of MinLP by relating it to a subproblem of set cover. Then, we define a new graph operation, called label squaring, and use it to derive a self-improvability property. Finally, we establish the main result by exploiting this property and additional structure common to instances obtained from the reduction described in the first stage.

Lemma 7. *There exists a constant $c > 0$, such that a polynomial time algorithm approximating MinLP within a factor of $c \ln n$ implies* P = NP.

Proof. By plugging the proof system of Raz and Safra [19] (or alternatively, Arora and Sudan [3]) into the reduction of Bellare, Goldwasser, Lund and Russell [5], the former authors showed that set cover is NP-hard to approximate within a factor of $O(\log n)$. In other words, there is a constant $c' > 0$ such that approximating set cover in polynomial time within a factor of $c' \ln n$ implies P = NP. This result also applies to instances $(U, \mathcal{S})$ with $|U| > |\mathcal{S}|^{1/q}$, for some constant $q \geq 1$, since the above-mentioned construction guarantees that $|U|$ and $|\mathcal{S}|$ are polynomially related [2]. We refer to this special case as MinSC′.

Given a MinSC′ instance, consisting of a ground set $U = \{e_1, \ldots, e_n\}$ and a family of subsets $\mathcal{S} = \{S_1, \ldots, S_m\} \subseteq 2^U$, we define an instance $(G, \mathcal{L}, s, t)$ of MinLP as follows:

1. The vertices of G are $v_0, \ldots, v_n$. In addition, for each element $e_j \in U$ we create a gadget $G(e_j)$ by connecting v_{j-1} and v_j to the upper rung of a ladder graph. More precisely, if e_j belongs to p subsets in $\mathcal{S}$, we put together a ladder whose rungs are $(a_j^1, b_j^1), \ldots, (a_j^p, b_j^p)$, adding the edges (v_{j-1}, a_j^1) and (v_j, b_j^1). This configuration is illustrated in Figure 4.

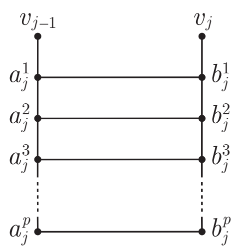

Fig. 4. The gadget $G(e_j)$

2. We now spread the labels $\{\ell_0, \ell_1, \ldots, \ell_m\}$ on the edges of G. Using the notation of item 1, each of the p rungs is given a distinct label from $\{\ell_i : e_j \in S_i\}$, whereas all other edges of $G(e_j)$ are given the label ℓ_0.
3. We set $s = v_0$ and $t = v_n$.

At this point, it is imperative to remark that since $n > m^{1/q}$, the above construction ensures that the overall number of vertices satisfies

$$|V(G)| \leq n + 1 + 2nm \leq n + 1 + 2n^{q+1} \leq 4n^{q+1} \ .$$

Now let $c = \frac{c'}{4(q+1)}$, and suppose that MinLP can be approximated in polynomial time within a factor of $c \ln |V(G)|$. We show that this assumption leads to an approximation factor of at most $c' \ln n$ for MinSC′, implying P = NP. To this end, let $\mathcal{S}^* \subseteq \mathcal{S}$ be an optimal solution to the instance $(U, \mathcal{S})$. As all elements of U are covered by $\mathcal{S}^*$, the label of at least one rung in each of the n ladders belongs to $\{\ell_i : S_i \in \mathcal{S}^*\}$, and by augmenting this label set with ℓ_0 we obtain a subgraph of G that contains an s-t path. Therefore, the number of labels in an optimal solution to the new MinLP instance is at most $|\mathcal{S}^*| + 1$. It follows that we can find in polynomial time an s-t path P satisfying

$$\begin{aligned} |\mathcal{L}(P)| &\leq \frac{c'}{4(q+1)} \ln |V(G)| \cdot (|\mathcal{S}^*| + 1) \leq \frac{c'}{4(q+1)} \ln \left(4n^{q+1}\right) \cdot 2|\mathcal{S}^*| \\ &\leq \frac{c'}{2} \ln (4n) \cdot |\mathcal{S}^*| \leq c' \ln n \cdot |\mathcal{S}^*| \ . \end{aligned}$$

The second inequality holds since $|V(G)| \leq 4n^{q+1}$, and the last inequality follows from observing that we can assume without loss of generality that $n \geq 4$, so $\ln(4n) \leq 2 \ln n$. It is not difficult to verify that, as the path P necessarily traverses ℓ_0-labeled edges, $\{S_i \in \mathcal{S} : \ell_i \in \mathcal{L}(P)\}$ is a cover of U with cardinality at most $|\mathcal{L}(P)| - 1 \leq c' \ln n \cdot |\mathcal{S}^*|$.

□

Given a MinLP instance $I = (G, \mathcal{L}, s, t)$, its *label squaring* $I^2 = (G^2, \mathcal{L}^2, s^2, t^2)$ is a new instance defined as follows. To assemble the graph G^2, we first construct a distinct copy G_e of G for each original edge $e \in E(G)$. Letting s_e and t_e denote the vertices of G_e that correspond to s and t, we arbitrarily assign s_e and t_e to different endpoints of e. Then, for each $v \in V(G)$, the vertices assigned to v are unified, over all copies, to a single vertex v^2. Using this notation, the source and destination are s^2 and t^2, respectively. Finally, the new set of labels is $\mathcal{L}(G) \times \mathcal{L}(G)$, where the edge of G_e corresponding to $f \in E(G)$ is given the label $(\mathcal{L}(e), \mathcal{L}(f))$.

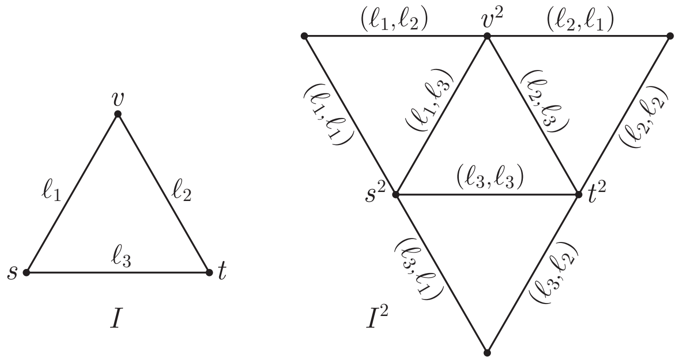

Fig. 5. A label squaring example

Lemma 8. $\mathrm{OPT}(I^2) \le \mathrm{OPT}^2(I)$.

Lemma 9. *There is a polynomial-time algorithm that, given an s^2-t^2 path P^2 in G^2, finds an s-t path P in G satisfying $|\mathcal{L}(P)| \le |\mathcal{L}^2(P^2)|^{1/2}$.*

Theorem 10. *For any fixed $k \ge 1$, MinLP cannot be approximated in polynomial time within a factor of $O(\log^k n)$ unless* P = NP.

Proof. The reduction described in Lemma 7 produces MinLP instances in which the underlying graph is planar. Therefore, the result stated in this lemma also applies to instances $I = (G, \mathcal{L}, s, t)$ in which G is an n-vertex planar graph. Now suppose that there exists a polynomial-time algorithm $\mathcal{A}$ whose approximation factor for such instances is $\alpha(n) \le c_k \ln^k n$, for some $c_k > 0$. We show that this algorithm can utilize the label squaring operation to obtain a self-improvability property, as a result of which we derive an approximation factor smaller than $c \ln n$ for planar MinLP, where c is the constant mentioned in Lemma 7.

We assume that n is sufficiently large so that $\ln^{1/2}(3n) \le \frac{c}{4} \ln n$, and let $q = q(k, c_k)$ be the smallest integer satisfying $c_k^{2^{-q}} \le 2$, $2^{2^{-q}qk} \le 2$ and $2^{-q}k \le \frac{1}{2}$. Such a constant indeed exists, since $c_k^{2^{-q}} \to 1$, $2^{2^{-q}qk} \to 1$, and $2^{-q}k \to 0$ as q tends to infinity. Starting with a planar instance I, we repeatedly apply the label squaring operation q times, to obtain $I^{2^q} = (G^{2^q}, \mathcal{L}^{2^q}, s^{2^q}, t^{2^q})$. We then employ the algorithm $\mathcal{A}$ to find an approximate s^{2^q}-t^{2^q} path in G^{2^q}, and make use of Lemmas 8 and 9 to obtain an s-t path P in G such that

$$|\mathcal{L}(P)| \le \left(\alpha\left(|V(G^{2^q})|\right) \cdot \mathrm{OPT}\left(I^{2^q}\right)\right)^{2^{-q}} \le \alpha^{2^{-q}}\left(|V(G^{2^q})|\right) \cdot \mathrm{OPT}(I) \ .$$

To bound the approximation guarantee $\alpha^{2^{-q}}(|V(G^{2^q})|)$ in terms of n and c, we first claim that the number of vertices in G^{2^q} is at most $(3n)^{2^q}$. For this purpose, it can be easily verified that the label squaring operation preserves planarity, implying that the instances I^{2^j} we obtain throughout the sequence are planar, and in particular $|E(G^{2^j})| \le 3|V(G^{2^j})| - 6$. By combining this property with the observation that $|V(G^{2^{j+1}})| \le |V(G^{2^j})| \cdot |E(G^{2^j})|$, we can inductively prove that $|E(G^{2^j})| \le (3n)^{2^j}$ and $|V(G^{2^j})| \le 3^{2^j - 1} n^{2^j}$, with room to spare. It follows that the approximation factor we derive is at most

$$\alpha^{2^{-q}}\left(|V(G^{2^q})|\right) \le c_k^{2^{-q}} \ln^{2^{-q}k}\left((3n)^{2^q}\right) = c_k^{2^{-q}} 2^{2^{-q}qk} \ln^{2^{-q}k}(3n) \le 4 \ln^{1/2}(3n) \le c \ln n \ .$$

□

References

1. A. A. Ageev and M. Sviridenko. Pipage rounding: A new method of constructing algorithms with proven performance guarantee. *Journal of Combinatorial Optimization*, 8(3):307–328, 2004.
2. S. Arora. Personal communication, November 2005.
3. S. Arora and M. Sudan. Improved low-degree testing and its applications. *Combinatorica*, 23(3):365–426, 2003.
4. A. Avidor and U. Zwick. Approximating MIN 2-SAT and MIN 3-SAT. *Theory of Computing Systems*, 38(3):329–345, 2005.

5. M. Bellare, S. Goldwasser, C. Lund, and A. Russell. Efficient probabilistically checkable proofs and applications to approximations. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, pages 294–304, 1993.
6. H. Broersma, X. Li, G. Woeginger, and S. Zhang. Paths and cycles in colored graphs. *Australasian Journal on Combinatorics*, 31:299–311, 2005.
7. T. Brüggemann, J. Monnot, and G. J. Woeginger. Local search for the minimum label spanning tree problem with bounded color classes. *Operations Research Letters*, 31(3):195–201, 2003.
8. R. D. Carr, S. Doddi, G. Konjevod, and M. V. Marathe. On the red-blue set cover problem. In *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 345–353, 2000.
9. R.-S. Chang and S.-J. Leu. The minimum labeling spanning trees. *Information Processing Letters*, 63(5):277–282, 1997.
10. I. Dinur and S. Safra. On the hardness of approximating minimum vertex cover. *Annals of Mathematics*, 162(1):439–486, 2005.
11. O. Goldschmidt, D. S. Hochbaum, and G. Yu. A modified greedy heuristic for the set covering problem with improved worst case bound. *Information Processing Letters*, 48(6):305–310, 1993.
12. R. Hassin. Approximation schemes for the restricted shortest path problem. *Mathematics of Operations Research*, 17(1):36–42, 1992.
13. R. Hassin, J. Monnot, and D. Segev. Approximation algorithms and hardness results for labeled connectivity problems, 2006. Available at `http://www.math.tau.ac.il/~segevd`.
14. D. R. Karger, R. Motwani, and G. D. S. Ramkumar. On approximating the longest path in a graph. *Algorithmica*, 18(1):82–98, 1997.
15. S. Khuller, A. Moss, and J. Naor. The budgeted maximum coverage problem. *Information Processing Letters*, 70(1):39–45, 1999.
16. S. O. Krumke and H.-C. Wirth. On the minimum label spanning tree problem. *Information Processing Letters*, 66(2):81–85, 1998.
17. D. H. Lorenz and D. Raz. A simple efficient approximation scheme for the restricted shortest path problem. *Operations Research Letters*, 28(5):213–219, 2001.
18. M. V. Marathe and S. S. Ravi. On approximation algorithms for the minimum satisfiability problem. *Information Processing Letters*, 58(1):23–29, 1996.
19. R. Raz and S. Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, pages 475–484, 1997.
20. A. Srinivasan. Distributions on level-sets with applications to approximation algorithms. In *Proceedings of the 42nd Annual Symposium on Foundations of Computer Science*, pages 588–597, 2001.
21. Y. Wan, G. Chen, and Y. Xu. A note on the minimum label spanning tree. *Information Processing Letters*, 84(2):99–101, 2002.
22. H.-C. Wirth. *Multicriteria Approximation of Network Design and Network Upgrade Problems*. PhD thesis, Department of Computer Science, Würzburg University, 2001.
23. Y. Xiong, B. Golden, and E. Wasil. Worst-case behavior of the MVCA heuristic for the minimum labeling spanning tree problem. *Operations Research Letters*, 33(1):77–80, 2005.