

Algorithmic Aspects of the S -Labeling Problem

Guillaume Fertin¹, Irena Rusu¹, and Stéphane Vialette²(✉)

¹ LINA UMR CNRS 6241, Université de Nantes, Nantes, France

{guillaume.fertin,irena.rusu}@univ-nantes.fr

² LIGM CNRS UMR 8049, Université Paris-Est, Champs-sur-Marne, France

vialette@univ-mlv.fr

Abstract. Given a graph $G = (V, E)$ of order n and maximum degree Δ , the **NP**-complete S -LABELING problem consists in finding a labeling of G , *i.e.* a bijective mapping $\phi : V \rightarrow \{1, 2, \dots, n\}$, such that $\text{SL}_\phi(G) = \sum_{\{u,v\} \in E} \min\{\phi(u), \phi(v)\}$ is minimized. A preliminary study of the S -LABELING problem has been undertaken in [9]; here, we prolongate this study, and focus more specifically on algorithmic results concerning the problem. We first give intrinsic properties of optimal labelings, which will prove useful for our algorithmic study. We then show that the S -LABELING problem is polynomial-time solvable for (sets of) caterpillars. We also provide upper and lower bounds on $\text{SL}_\phi(G)$, that in turn allow us to determine polynomial-time approximation algorithms for different classes of graphs such as regular graphs, connected graphs and forests, but also for general graphs. Concerning exact algorithms, we show that the problem is solvable in $O^*(1.44225^{n-\Delta})$ time, and that deciding whether there exist a labeling ϕ of G such that $\text{SL}_\phi(G) \leq |E| + k$ is solvable in $O^*(2^{2\sqrt{k}} (2\sqrt{k})!)$.

1 Introduction

A *labeling* of a graph G is an assignment of distinct integers to its vertices, in such a way that a certain objective function is optimized. In the literature, a graph labeling of G is also called a *linear arrangement*, a *linear layout* or a *numbering* of the vertices of G . A large amount of relevant combinatorial problems in different areas can be rephrased as graph labeling problems, which have been widely investigated during the last decades. These include numerical analysis [17], VLSI circuit design [4], network reliability [15], computational biology [16], scheduling [1], parallel processing [18], etc. However, for most objective functions, the derived graph labeling problem turns out to be **NP**-complete. Popular graph labeling problems arising in the above-mentioned applications include BANDWIDTH [10], MINIMUM LINEAR ARRANGEMENT [13], CUTWIDTH [2] and VERTEX SEPARATION [19]. There exist several surveys that deal with different aspects of graph labeling problems [3, 5, 8].

In this paper, we are interested in the S -LABELING problem [9, 22], which, given a graph $G = (V, E)$, asks for a graph labeling $\phi : V \rightarrow \{1, 2, \dots, |V|\}$ such that the total sum $\text{SL}_\phi(G) = \sum_{\{u,v\} \in E} \min\{\phi(u), \phi(v)\}$ is minimized, and we write $\text{SL}(G)$ for the minimum $\text{SL}_\phi(G)$ over all possible labelings of G . Note

that, in this problem (and thus, all along this paper), we do not require G to be connected. However, we can always assume G has no isolated vertices. Indeed, if this was the case, let V_I be the set of isolated vertices in G . Since in any labeling ϕ of G , no vertex from V_I contributes to $\text{SL}_\phi(G)$, we can always label the vertices with the highest possible labels from V_I , or equivalently, change G into $G' = (V \setminus V_I, E)$.

The S -LABELING problem has been proved to be **NP**-complete for planar at most cubic graphs [22], but its complexity is unknown for trees, forests, and more generally, for bipartite graphs. The problem has also been studied in [9], where $\text{SL}(G)$ is determined for classical classes of graphs (paths, cycles, complete (bipartite) graphs). Upper and lower bounds are also given for general graphs, and approximation ratios are given for trees, regular graphs, and general graphs. The S -LABELING problem has also been proved to be polynomial-time solvable for split graphs [9].

This paper can be seen as a (late) follow-up of [9], mainly focused on algorithmic aspects of the S -LABELING problem, and is organized as follows. Section 2 introduces terminology used throughout this paper. Section 3 presents essential properties of optimal labelings. In Sect. 4, we prove that the S -LABELING problem is polynomial-time solvable for (sets of) caterpillars. In Sect. 5, we investigate polynomial-time approximation and exponential-time algorithms. Due to space constraints, most proofs are omitted and deferred to the full version of this paper.

2 Notations

A *graph* is an ordered pair $G = (V, E)$ comprising a set V of vertices together with a set E of edges, which are 2-element subsets of V . The *order* n (resp. *size* m) of G is its number of vertices (resp. edges). The degree of a vertex $u \in V$, denoted $d_G(u)$ (or $d(u)$ if clear from the context), is the number of edges that are incident to u . We write $\Delta(G)$ (or Δ , if clear from the context) for the maximum degree of G . A *vertex cover* of $G = (V, E)$ is a subset of vertices $V' \subseteq V$ such that each edge $e \in E$ is incident to at least one vertex of V' . The size of a minimum cardinality vertex cover of G is denoted $\tau(G)$ (or τ , if clear from the context). An *independent set* of $G = (V, E)$ is a subset of vertices $V' \subseteq V$, no two of which are adjacent. A *tree* is a graph in which any two vertices are connected by exactly one simple path, and a *caterpillar* is a tree in which all the vertices are within distance 1 of a central path (equivalently, caterpillars are the trees in which there exists a path that contains every node of degree two or more). A *forest* is a collection of trees.

Let $G = (V, E)$ be a graph of order n . A *labeling* of G is a bijective mapping $\phi : V \rightarrow \{1, 2, \dots, n\}$, and we denote by $\Phi(G)$ the set of all labelings of G . The *S -labeling number* of G with respect to some labeling $\phi \in \Phi(G)$, denoted $\text{SL}_\phi(G)$, is defined to be $\text{SL}_\phi(G) = \sum_{\{u,v\} \in E} \min\{\phi(u), \phi(v)\}$. To abbreviate notations, we write $\text{SL}(G) = \min\{\text{SL}_\phi(G) : \phi \in \Phi(G)\}$ and we let $\Phi_{\text{opt}}(G) \subseteq \Phi(G)$ stand for the set of all optimal labelings (*i.e.*, $\Phi_{\text{opt}}(G) = \{\phi \in \Phi(G) : \text{SL}_\phi(G) = \text{SL}(G)\}$). Let $\phi \in \Phi(G)$ be a labeling of G .

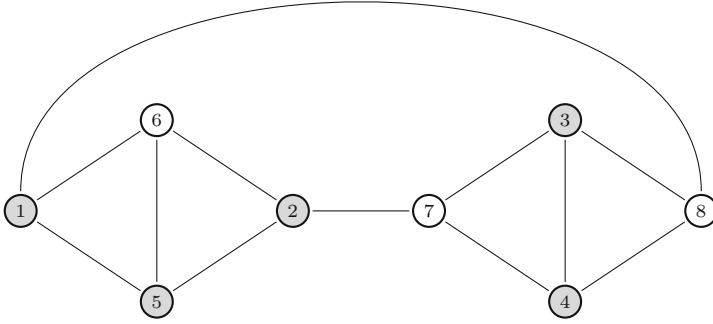


Fig. 1. A graph G together with a labeling ϕ . Vertices in V_ϕ^+ are in grey. We have $c_\phi(\phi^{-1}(1)) = c_\phi(\phi^{-1}(2)) = c_\phi(\phi^{-1}(3)) = 3$, $c_\phi(\phi^{-1}(4)) = 2$, $c_\phi(\phi^{-1}(5)) = 1$, and $c_\phi(\phi^{-1}(6)) = c_\phi(\phi^{-1}(7)) = c_\phi(\phi^{-1}(8)) = 0$, yielding $SL_\phi(G) = (3 \times 1) + (3 \times 2) + (3 \times 3) + (2 \times 4) + (1 \times 5) + (0 \times 6) + (0 \times 7) + (0 \times 8) = 31$.

For any vertex $u \in V$, the *contribution* $c_\phi(u)$ of u to the S -labeling number SL_ϕ is the integer defined as $c_\phi(u) = |\{v \in V : \{u, v\} \in E \text{ and } \phi(u) < \phi(v)\}|$. We will be mostly interested, in the rest of the paper, in those vertices with non-zero contribution. To this aim, we define $V_\phi^+ = \{u \in V : c_\phi(u) > 0\}$. An example illustrating these notions is given in Fig. 1.

3 Properties of Optimal Labelings

Towards investigating computational issues, it is critical to support a deeper understanding of the structure of optimal labelings. This is the purpose of the current section.

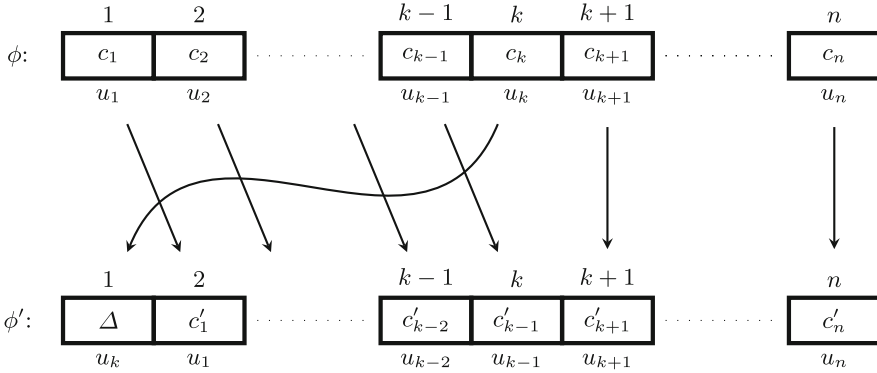
Property 1. For any graph $G = (V, E)$ and any labeling $\phi \in \Phi(G)$: (a) V_ϕ^+ is a vertex cover of G , (b) $\sum_{u \in V_\phi^+} c_\phi(u) = |E|$, and (c) $SL_\phi(G) = \sum_{u \in V_\phi^+} c_\phi(u) \phi(u)$.

Lemma 1. [9] *Let $G = (V, E)$ be a graph, and $\phi \in \Phi_{\text{opt}}(G)$. For any two vertices $u, v \in V$, if $\phi(u) < \phi(v)$ then $c_\phi(u) \geq c_\phi(v)$.*

Lemma 2. *Let $G = (V, E)$ be a graph, and $\phi \in \Phi_{\text{opt}}(G)$. For any distinct vertices $u, v \in V$, if $c_\phi(u) = c_\phi(v)$ then $\{u, v\} \notin E$.*

We now turn to proving that, in any optimal labeling ϕ of G , the vertex ranked first by ϕ is a maximum degree vertex.

Lemma 3. *Let $G = (V, E)$ be a graph, let $\phi \in \Phi_{\text{opt}}(G)$, and let $u \in V$ be the vertex satisfying $\phi(u) = 1$. Then $c_\phi(u) = \Delta(G)$.*



Proof. Let $G = (V, E)$ be a graph, and $\phi \in \Phi_{\text{opt}}(G)$ be an optimal labeling of G . Let $k = \min\{\phi(u) : u \in V \wedge c_\phi(u) = \Delta(G)\}$. We show that $k = 1$ thereby proving the lemma. Write Δ for $\Delta(G)$. For convenience, for $1 \leq i \leq n$, let u_i and c_i stand for $\phi^{-1}(i)$ and $c_\phi(\phi^{-1}(i)) = c_\phi(u_i)$, respectively.

Suppose, aiming at a contradiction, that $k > 1$. Consider the labeling ϕ' of G defined as follows: $\phi'(u_k) = 1$, $\phi'(u_i) = \phi(u_i) + 1$ for any $1 \leq i \leq k - 1$ and $\phi'(u_i) = \phi(u_i)$ for any $k + 1 \leq i \leq n$. For convenience, for $1 \leq i \leq n$, $i \neq k$, let c'_i stand for $c_{\phi'}(u_i)$ (see above figure). We thus have $\text{SL}_\phi(G) = \sum_{i=1}^n ic_i$ and $\text{SL}_{\phi'}(G) = \Delta + \sum_{i=1}^{k-1} (i + 1)c'_i + \sum_{i=k+1}^n ic_i$. Now, if we let D stand for $\text{SL}_\phi(G) - \text{SL}_{\phi'}(G)$, we obtain $D = \sum_{i=1}^n ic_i - \Delta - \sum_{i=1}^{k-1} (i + 1)c'_i - \sum_{i=k+1}^n ic_i = \sum_{i=1}^k ic_i - \Delta - \sum_{i=1}^{k-1} (i + 1)c'_i$ (Eq. 1). We need the following inequality.

Claim 1. $\sum_{i=1}^{k-1} (i + 1)c'_i \leq \sum_{i=1}^{\Delta - c_k} (i + 1)(c_i - 1) + \sum_{i=\Delta - c_k + 1}^{k-1} (i + 1)c_i$.

Proof. By construction, $c_i - 1 \leq c'_i \leq c_i$ for $1 \leq i \leq k - 1$. Moreover, since exactly $\Delta - c_k$ vertices of $\{u_1, u_2, \dots, u_{k-1}\}$ are connected with vertex u_k , then there exist $S \subseteq \{1, 2, \dots, k - 1\}$ of size $\Delta - c_k$ such that $c'_i = c_i - 1$ if and only if $i \in S$. If we let \bar{S} stand for $\{1, 2, \dots, k - 1\} \setminus S$, then $\sum_{i=1}^{k-1} (i + 1)c'_i = \sum_{i \in S} (i + 1)(c_i - 1) + \sum_{i \in \bar{S}} (i + 1)c_i = \sum_{i=1}^{k-1} (i + 1)c_i - \sum_{i \in S} (i + 1)$. This latter sum is certainly maximized for $S = \{1, 2, \dots, \Delta - c_k\}$, and hence $\sum_{i=1}^{k-1} (i + 1)c'_i \leq \sum_{i=1}^{\Delta - c_k} (i + 1)(c_i - 1) + \sum_{i=\Delta - c_k + 1}^{k-1} (i + 1)c_i$. \square

Combining Claim 1 with (Eq. 1) yields $D \geq \sum_{i=1}^k ic_i - \Delta - \sum_{i=1}^{\Delta - c_k} (i + 1)(c_i - 1) - \sum_{i=\Delta - c_k + 1}^{k-1} (i + 1)c_i = \sum_{i=1}^k ic_i - \Delta - \sum_{i=1}^{k-1} ic_i - \sum_{i=1}^{k-1} c_i + \sum_{i=1}^{\Delta - c_k} (i + 1) = kc_k - \Delta - \sum_{i=1}^{k-1} c_i + \sum_{i=1}^{\Delta - c_k} (i + 1) = kc_k - \Delta - \sum_{i=1}^{k-1} c_i + \frac{(\Delta - c_k + 3)(\Delta - c_k)}{2}$.

But $c_i \leq \Delta - 1$ for $1 \leq i \leq k - 1$ (this follows from $k > 1$ and Lemma 1), and hence $\sum_{i=1}^{k-1} c_i \leq (k - 1)(\Delta - 1)$. Then it follows that $D \geq kc_k - \Delta - (k - 1)(\Delta - 1) + \frac{(\Delta - c_k + 3)(\Delta - c_k)}{2} = (k - 1) - k(\Delta - c_k) + \frac{(\Delta - c_k + 3)(\Delta - c_k)}{2} = (k - 1) + \frac{(\Delta - c_k)(\Delta - c_k - 2k + 3)}{2}$. Combining the above with $\Delta - c_k \geq 1$ yields $D \geq (k - 1) + \frac{4 - 2k}{2} \geq 1$, and hence $\text{SL}_\phi(G) > \text{SL}_{\phi'}(G)$. This is the desired contradiction since ϕ is an optimal labeling of G . \square

Lemma 4. *Let $G = (V, E)$ be a graph, $\phi \in \Phi_{\text{opt}}(G)$, and let ϕ' be the labeling obtained from ϕ by swapping the labels of any two vertices $u, v \in V$ such that $c_\phi(u) = c_\phi(v)$. Then $\phi' \in \Phi_{\text{opt}}(G)$.*

Lemma 5. *Let $G = (V, E)$ be a graph, $\phi \in \Phi_{\text{opt}}(G)$, and let $u, v \in V$ such that $\{u, v\} \in E$ and $c_\phi(u) = c_\phi(v) - 1$. Then an optimal labeling ϕ' with $\phi'(u) = \phi'(v) - 1$ may be obtained from ϕ by a series of label swaps involving only vertices $z \in V$ with $c_\phi(z) \in \{c_\phi(u), c_\phi(v)\}$ and $\phi(v) < \phi(z) < \phi(u)$.*

Coming back to V_ϕ^+ , and as discussed in [9], in the light of Property 1(a), it would be tempting to claim that $\tau(G) = |V_\phi^+|$ for any –or at least one– optimal labeling $\phi \in \Phi_{\text{opt}}(G)$. Unfortunately, this is not true: there exist a graph G for which any optimal labeling ϕ satisfies $\frac{|V_\phi^+|}{\tau(G)} = \frac{5}{4}$ [9]. This raises the following question: for any graph G , does there exist a labeling $\phi \in \Phi_{\text{opt}}(G)$ such that $\frac{|V_\phi^+|}{\tau(G)} = O(1)$? This question remains open. However, we have the following result, which improves Lemma 1.3 from [9] by a factor $\sqrt{2}$.

Lemma 6. *For any graph G of maximum degree Δ , there exist an optimal labeling $\phi \in \Phi_{\text{opt}}(G)$ for which $\frac{|V_\phi^+|}{\tau(G)} < \sqrt{\Delta}$.*

4 A Polynomial-Time Algorithm for Sets of Caterpillars

In this section, we prove that the S -LABELING problem is in \mathbf{P} when the input instance is a (set of) caterpillar(s) (see Proposition 1). This result can be seen as a step towards understanding the complexity of the problem in trees and forests, which remains unknown. Recall that a caterpillar is a tree $G = (V, E)$ for which all the vertices are within distance 1 of a dominating path. It is easy to see that every longest path P of G has the property that, for all $v \in V$, either v belongs to P or v is a leaf adjacent to a vertex of P , which is not an endpoint of P . We call *linear representation* of G , denoted $\text{LR}(G)$, any drawing of G in which the vertices of some longest path P of G lie along an horizontal line according to their order on the path. We call *set of caterpillars* any vertex-disjoint set of caterpillars. A set of caterpillars is seen as a graph $G = (V, E)$ whose connected components $C_1, C_2 \dots C_p$ are caterpillars. The linear representation of G , still denoted $\text{LR}(G)$, is in this case the sequence $\text{LR}(C_1), \text{LR}(C_2) \dots \text{LR}(C_p)$ of the linear representations of its connected components, successively on the same horizontal line. Recall that G being a set of caterpillars on n vertices, G contains $O(n)$ edges.

Lemma 7. *Let G be a set of caterpillars, and u be its leftmost vertex in $\text{LR}(G)$ such that $d_G(u) = \Delta$. There is an optimal labeling ϕ_{opt} of G such that $\phi_{\text{opt}}(u) = 1$.*

Algorithm 1. Greedy computation of a labeling of a set of caterpillars.

Algorithm: GreedyCat**Data:** A set of caterpillars $G = (V, E)$ **Result:** A labeling ϕ of G $i \leftarrow 1$ **while** $G \neq \emptyset$ **do** Compute $\text{LR}(G)$ Let u be the leftmost vertex in $\text{LR}(G)$ with $d_G(u) = \Delta(G)$ $\phi(u) \leftarrow i$ $i \leftarrow i + 1$ $G \leftarrow G - u$ **return** (ϕ)

Proof. Notice that if we prove the existence of an optimal labeling $\phi \in \Phi_{\text{opt}}(G)$ such that $c_\phi(u) = \Delta$, then we are done. Indeed, by Lemma 3, we know that the vertex z with label 1 has contribution Δ , and by Lemma 4, we know that swapping the labels of u and z yields an optimal labeling ϕ_{opt} such that $\phi_{\text{opt}}(u) = 1$.

We now show that there is an optimal labeling ϕ of G such that $c_\phi(u) = \Delta$. Let ϕ' be an optimal labeling of G such that every leaf v of G that is adjacent to u (if any) satisfies $\phi'(v) > \phi'(u)$ (say this is a *regular* leaf). To prove that such a labeling exists, assume by contradiction that this is not the case, and let ϕ' be chosen so as to minimize the number of leaves with $\phi'(v) < \phi'(u)$. Denote v_0 the leaf with this property and such that $\phi'(v_0)$ is maximum. By Lemma 1, we know that $c_{\phi'}(u) \leq c_{\phi'}(v_0)$, with $c_{\phi'}(v_0) = 1$. We thus necessarily have $c_{\phi'}(u) = 0$ and $c_{\phi'}(v_0) = 1$ by Lemma 2, $\{u, v_0\}$ being an edge in G . In this case, Lemma 5 ensures that some labels of ϕ' may be swapped to ensure $\phi'(v_0) > \phi'(u)$, and that these swaps do not affect regular leaves. But this contradicts the initial choice of ϕ' . As a consequence, all the leaves adjacent to u are regular, and $c_{\phi'}(u) \geq \Delta - 2$ since at most two neighbors of u are not leaves.

Now, let v (respectively w) be the right (respectively left) neighbor of u , if it exists. Then, since u is the leftmost vertex in $\text{LR}(G)$ with degree Δ , we have $d_G(w) < \Delta$ and $d_G(v) \leq \Delta$. We transform ϕ' into an optimal labeling ϕ such that $\phi(u) < \phi(w)$ and $\phi(u) < \phi(v)$, in two steps. First, if $\phi'(u) < \phi'(w)$, then we define $\phi'' = \phi'$. Otherwise, we have $\phi'(u) > \phi'(w)$, and since u and w are adjacent, we deduce that $c_{\phi'}(u) < c_{\phi'}(w)$ by Lemmas 1 and 2. The remarks that $c_{\phi'}(u) \geq \Delta - 2$ and $c_{\phi'}(w) \leq \Delta - 1$ further imply that we necessarily have $c_{\phi'}(u) = \Delta - 2$ and $c_{\phi'}(w) = \Delta - 1$, thus allowing us to apply Lemma 5 in order to deduce the existence of an optimal labeling ϕ'' with $\phi''(u) < \phi''(w)$. Second, if $\phi''(u) < \phi''(v)$, then we define $\phi = \phi''$. Otherwise, since u and v are adjacent, we deduce that $c_{\phi''}(u) < c_{\phi''}(v)$ by Lemmas 1 and 2. The remarks that $c_{\phi''}(u) \geq \Delta - 1$ (we know that $\phi''(w) > \phi''(u)$) and $c_{\phi''}(v) \leq \Delta$ further imply that we necessarily have $c_{\phi''}(u) = \Delta - 1$ and $c_{\phi''}(v) = \Delta$, thus allowing us to apply Lemma 5 in order to deduce the existence of an optimal labeling ϕ with $\phi(u) < \phi(v)$. Notice that the label changes performed according to Lemma 5 do not affect the label of w . Then, $\phi(u) < \phi(x)$ for all the neighbors x of u , and thus $c_\phi(u) = \Delta$. \square

Proposition 1. *Algorithm **GreedyCat** optimally solves the S -LABELING problem for any set of caterpillars of order n , in $O(n \log n)$ time.*

Proof. By contradiction, assume the labeling ϕ obtained by **GreedyCat** is not an optimal labeling. Let $v_1, v_2 \dots v_n$ be the vertices of G ordered such that $\phi(v_i) = i$. Now, consider an optimal labeling ϕ_{opt} which maximizes the value k with the property that $\phi_{\text{opt}}(v_i) = \phi(v_i) (= i)$ for all $i, 1 \leq i \leq k$. Let G_{k+1} be the subgraph induced by $\{v_{k+1} \dots v_n\}$ in G .

According to **GreedyCat**, $v_1 \dots v_n$ are labeled successively in this order, and when v_{k+1} is labeled, it has maximum degree in G_{k+1} and is the leftmost in $\text{LR}(G)$ with this property. It is easy to note that a labeling ϕ_0 of G with $\phi_0(v_i) = i$ for all $i, 1 \leq i \leq k$, is optimal for G if and only if the labeling ϕ_1 of G_{k+1} , defined by $\phi_1(v_i) = \phi_0(v_i) - k$ for $k + 1 \leq i \leq n$, is optimal for G_{k+1} . This is due to the constant cost of the edges $\{v_i, v_j\}$ with $i \leq k$ or $j \leq k$ within $\text{SL}(G)$. By Lemma 7, there is an optimal labeling ϕ_1 of G_{k+1} such that $\phi_1(v_{k+1}) = 1$. The resulting labeling $\phi_0(v_i) = \phi_1(v_i) + k$ for $k + 1 \leq i \leq n$, extended to G by $\phi_0(v_i) = i$ for $1 \leq i \leq k$, is then optimal and satisfies $\phi_0(v_i) = \phi(v_i) (= i)$ for all $i, 1 \leq i \leq k + 1$. This contradicts the choice of ϕ_{opt} .

Concerning the complexity, the initial computation of $\text{LR}(G)$ is done in $O(n)$ time by using traversals starting with a 1-degree vertex. The vertices occurring on the horizontal line of the representation are then renumbered in increasing order from left to right, whereas the leaves receive the remaining numbers. Then, for each $d = 0, 1, 2 \dots \Delta$, we store all vertices of degree d in a balanced BST denoted $B[d]$. We furthermore need another balanced BST to store all the degrees d for which $B[d]$ is non-empty. It is easy to see that initializing all the BSTs is done in global time of $O(n \log n)$ by $n + \Delta$ insertions taking $O(\log n)$ time each. The update of the data structures we use involves, at each step, only the neighbors of the vertex u removed from G . There are $O(d_G(u))$ such neighbors, and the operations take no more than $O(\log n)$ time for each neighbor, which yields the required complexity. \square

5 Algorithmic Issues

This section contains two parts: the first one is devoted to approximating the S -LABELING problem, whereas the second part focuses on exact (*i.e.* exponential-time algorithms).

Approximating the S -labeling Problem. We begin by giving accurate lower and upper bounds of optimal labelings. For completeness, in Lemma 8, we recall the lower bound given in [9].

Lemma 8 [9]. *For any graph G of order n , size m and maximum degree Δ , $\text{SL}(G) \geq (m - \frac{\Delta}{2} \lfloor \frac{m}{\Delta} \rfloor) (\lfloor \frac{m}{\Delta} \rfloor + 1)$.*

Another general lower bound is given in Lemma 9 below.

Lemma 9. *For any graph G of size m , let ϕ be any labeling of G , and $|V_\phi^+| = k$. Then $SL_\phi(G) \geq \frac{k^2 - k + 2m}{2}$.*

Proof. Let ϕ be any labeling of G , and let $|V_\phi^+| = k$. Since every vertex in V_ϕ^+ has a strictly positive contribution (by definition), summing these over the k vertices in V_ϕ^+ adds up to $\frac{k(k+1)}{2}$, and covers k edges. For the remaining $m - k$ edges in G , the best scenario is that they all are incident to the vertex having label 1, which adds up to $m - k$. Hence the result. \square

We now turn to giving upper bounds for optimal labelings. The first two upper bounds are achieved by the following generic greedy algorithm, that we call **GreedyGen**: while G has edges, take a vertex u of maximum degree in G , assign the smallest available label to u (starting from 1), and remove u and its incident edges from G . Notice that a randomized algorithm achieving the same general upper bound as in Lemma 10 was given in [9]. Here, we improve this previous result by providing a *deterministic* algorithm, achieving the same performances for general graphs, and improving them for acyclic graphs.

Lemma 10. *For any graph G of order n and size m , Algorithm **GreedyGen** computes, in $O(m \log n)$ time, a labeling ϕ such that $SL_\phi(G) \leq \frac{m(n+1)}{3}$. Furthermore, if G is acyclic with $\Delta \geq 3$, ϕ satisfies $SL_\phi(G) \leq \frac{m(n+1)}{4}$.*

As a side remark, we observe that there exist classes of graphs for which each of the previously given upper and lower bounds are reached. Indeed, for any even n , $SL(C_n) = \frac{n^2 + 2n}{4} = \frac{\Delta \lfloor \frac{n}{\Delta} \rfloor (\lfloor \frac{n}{\Delta} \rfloor + 1)}{2}$; for any $k \leq m$, $SL(K_{1, m-k+1} \cup (k-1)K_2) = \frac{k^2 - k + 2m}{2}$; $SL(K_n) = \frac{1}{6}n(n^2 - 1) = \frac{m(n+1)}{3}$; and, finally, for any odd n , $SL(P_n) = \frac{m(n+1)}{4}$. Another general upper bound can be obtained by starting from a vertex cover V' of G , and applying a greedy strategy, similar to the one of Algorithm **GreedyGen**, but only taking into account vertices in V' . Since all vertices in $V \setminus V'$ have a zero contribution, they can be arbitrarily labeled from $|V'| + 1$ to n .

Lemma 11. *For any graph G of order n and size m , let V' be a vertex cover of G , with $|V'| = p$. A labeling $\phi_{V'}$ satisfying $SL_{\phi_{V'}}(G) \leq \frac{1}{2}m(p + 1)$ can be computed in $O(m \log p)$ time.*

We are now ready to state our results concerning approximation algorithms, which essentially rely on using the proper upper and lower bounds among the ones presented above. Some of these results improve or generalize the ones from [9], others are new, and all are deterministic (as opposed to some results from [9]). If G is a graph of size m and order n , we denote by $d = \frac{2m}{n}$ its average degree. The first general result we have is the following.

Proposition 2. *For any graph, **GreedyGen** is a $\frac{4\Delta}{3d}$ -approximation algorithm for the S -LABELING problem.*

Since we can always assume that G has no isolated vertices, we always have $d \geq 1$, and thus for any graph, we now that a $\frac{4\Delta}{3}$ approximation algorithm exists. Concerning regular graphs, setting $d = \Delta$ in Proposition 2 yields the following corollary.

Corollary 1. *For any regular graph, **GreedyGen** is a $\frac{4}{3}$ -approximation algorithm for the S -LABELING problem.*

Now, if we restrict ourselves to specific classes of graphs, we are able to obtain better approximation ratios. Let $\mathcal{F}_{\leq C}$ (resp. $\mathcal{F}_{\geq C}$) be the set of forests having at most (resp. at least) C connected components.

Proposition 3. *For any graph G , **GreedyGen** is an approximation algorithm having the following ratios: (a) $\frac{\Delta}{2}$ if $G \in \mathcal{F}_{\leq \Delta-1}$, provided $\Delta \geq 3$; (b) Δ if $G \in \mathcal{F}_{\geq \Delta}$; (c) $\frac{2\Delta}{3}$ if G is connected.*

We now give one more approximation algorithm, in relation to the MINIMUM VERTEX COVER problem. Let $\mathcal{G}_{\text{VC}}^\alpha$ denote the class of graphs for which the cardinality of a minimum vertex cover can be approximated within ratio α . The following result concerns graphs belonging to that class.

Proposition 4. *For any graph G in $\mathcal{G}_{\text{VC}}^\alpha$, there exist an $\alpha\Delta$ -approximation algorithm for the S -LABELING problem.*

In particular, for all graphs for which a minimum vertex cover can be computed in polynomial-time, the S -LABELING problem can be approximated within ratio Δ .

Exact Algorithms. After having discussed approximation algorithms, we now turn to exact algorithms. We first concentrate on exponential-time algorithms.

Proposition 5. *For any graph G of order n and maximum degree Δ , the S -LABELING problem is solvable in time $O^*(1.44225^{n\Delta})$, and in time $O^*(1.41422^{n\Delta})$ if G is a tree.*

Let G be a graph and $\phi \in \Phi_{\text{opt}}(G)$ be an optimal labeling of G . For every positive integer $c \neq 0$, let $V_\phi^c = \{u : u \in V \wedge c_\phi(u) = c\}$, and, for any non-empty V_ϕ^c , let $l_{\min}^c = \min\{\phi(u) : u \in V_\phi^c\}$ be the smallest label given by ϕ among the vertices in G having contribution equal to c by ϕ . Because of Lemma 2, we have the following corollary.

Corollary 2. *For any graph G , let $\phi \in \Phi_{\text{opt}}(G)$ be an optimal labeling of G . Then V_ϕ^c is a maximal independent set on the vertices of maximum degree in the induced graph $G[\{v : \phi(v) \geq l_{\min}^c\}]$.*

Proof (of Proposition 5). We use a Δ -bounded search tree based algorithm. The root of the search tree is labeled by $(G, 0, \emptyset)$. We explore the tree as follows. For a node (H, i, L) , we consider all maximal independent sets on the maximum degree vertices of H . For each maximal independent set $V' = \{u_1, u_2, \dots, u_k\}$,

we get a child labeled $(H', i+k, L')$, where $H' = H \setminus V'$ and $L' = L \cup \{(u_j, i+j) : 1 \leq j \leq k\}$. A leaf in the search tree is labeled (H, i, L) for some edgeless graph H and it evaluates to $\sum_{\{u,v\} \in E} \min\{\phi(u), \phi(v)\}$, where $\phi \in \Phi(G)$ is obtained from L in the natural way, *i.e.*, for every $u \in V$, $\phi(u) = i$ if $(u, i) \in L$. The optimal labeling is the minimum evaluation of a leaf of the search tree.

Correctness of the algorithm follows from Corollary 2. What is left is to prove that the search tree has depth bounded by $\Delta(G)$. This is again a consequence of Corollary 2, since the maximum degree of the graphs strictly decreases during the exploration of the tree. The time complexity now follows from the following result from [14]: any graph of order n contains at most 1.44225^n maximal independent sets, and these maximal independent sets can be enumerated in $O^*(1.44225^n)$ time. Besides, if G is a tree, we know by [23] that the number of maximal independent sets G is upper bounded by $O((\sqrt{2})^n)$ (more precisely, the number is $2^{n/2-1} + 1$ if n is even, and $2^{(n-1)/2}$ if n is odd). \square

As a side remark concerning Corollary 2, note that it would be tempting to push ahead and replace *maximal* by *maximum* in this corollary. However, that is pushing the argument too far, even for trees. Indeed, consider the tree given Fig. 2 with 3 maximum degree vertices. For one, starting from the unique maximum independent set $\{u, w\}$ on degree $\Delta = 3$ vertices yields a labeling with total sum 34. For another, starting from the other maximal independent set on degree $\Delta = 3$ vertices $\{v\}$ yields a labeling with total sum 31.

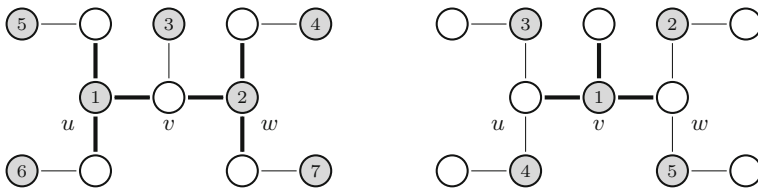


Fig. 2. A graph G with three maximum degree vertices u, v and w , each of degree $\Delta = 3$. (Left) An S -labeling ϕ_1 of G that starts with the unique maximum independent set on maximum degree vertices of cardinality 2 (namely, $\{u, w\}$), optimally completed with labels 3 to 8, yielding $SL_{\phi_1}(G) = 34$. (Right) An S -labeling ϕ_2 of G that starts with the unique maximal independent set on maximum degree vertices of cardinality 1 (namely, $\{v\}$), optimally completed with labels 3 to 8, yielding $SL_{\phi_2}(G) = 31 < SL_{\phi_1}(G)$.

Now, note that for any graph G of order n and size m , $SL(G) = \Omega(m)$. But since we can always assume G has no isolated vertices, $n \leq 2m$ and thus $SL(G) = \Omega(n)$. Therefore, the S -LABELING problem is fixed-parameter tractable [21] for any graph G , when the parameter is the size of the solution $SL(G)$. However, the above result relies on a parameter whose value can be considered as too high. Thus, we focus on the following problem: Given a graph $G = (V, E)$ of size $|E| = m$ and a positive integer k , is there a labeling $\phi \in \Phi(G)$ such that

$SL_\phi(G) \leq m + k$? We need the following result that gives a lower bound for $SL_\phi(G)$ in terms of $|V_\phi^+|$.

Lemma 12. *For any graph G of size m , let $\phi \in \Phi(G)$ be a labeling of G such that $SL_\phi(G) \leq m + k$ for some positive integer k . Then $|V_\phi^+| \leq 2\sqrt{k}$.*

Proof. Let $G = (V, E)$ be a graph and $\phi \in \Phi(G)$ be a labeling of G such that $SL_\phi(G) \leq |E| + k$ for some positive integer k . By Lemma 9, we know that $SL_\phi(G) \geq \frac{|V_\phi^+|^2 - |V_\phi^+| + 2m}{2}$, which yields $|V_\phi^+|^2 - |V_\phi^+| \leq 2k$. Solving this second degree inequality leads to $|V_\phi^+| \leq x$, where $x = \frac{1 + \sqrt{8k+1}}{2}$. A simple computation then shows that $x \leq 2\sqrt{k}$ for any positive integer k , which proves the result. \square

Proposition 6. *For any graph G of size m and any positive integer k , it can be decided in $O^*(2^{2\sqrt{k}} (2\sqrt{k})!)$ time whether $SL(G) \leq m + k$.*

In the above proposition, it is worth noticing that $k \geq \frac{m(n-4)}{4}$ for regular graphs (see Lemma 8), and hence Proposition 6 does not give fixed-parameter tractability for the above-guarantee parameterization variant [11, 20]. However, supposing G is of order n , we now determine how Proposition 6 compares to the brute-force $O^*(n!)$ time algorithm. First, we have $SL(G) \leq mn$, therefore $SL(G) \leq m + k$ implies $k \leq mn - m = m(n-1) \leq \frac{n(n-1)^2}{2}$, and hence $k = O(n^3)$. We now need the following lemma.

Lemma 13. *For positive x and n , if $2^x x! = n!$ then $x = \Theta(n)$.*

Substituting x by $2\sqrt{k}$ in Lemma 13 yields $k = \Theta(n^2)$. Since $2^x x!$ is an increasing function, we conclude that Proposition 6 improves on the brute-force $O^*(n!)$ time algorithm for any k , up to $\Theta(n^2)$.

6 Conclusion

We would like to end this paper with several open problems. First, what is the complexity of the S -LABELING problem for trees, forests, or more generally bipartite graphs? Second, does there exist a PTAS for the S -LABELING problem, for any graph G , or at least for specific classes of graphs? The same question holds for constant approximation ratios.

References

1. Adolphson, D.: Single machine job sequencing with precedence constraints. *SIAM J. Comput.* **6**, 40–54 (1977)
2. Adolphson, D., Hu, T.C.: Optimal linear ordering. *SIAM J. Appl. Math.* **25**(3), 403–423 (1973)
3. Bezrukov, S.: Edge isoperimetric problems on graphs. In: Lovasz, L., Gyárfás, A., Katona, B.O.H., Recski, A., Szekely, L. (eds.) *Graph Theory and Combinatorial Biology*, pp. 157–197. Janos Bolyai Mathematical Society, Budapest (1999)

4. Bhatt, S.N., Leighton, F.T.: A framework for solving VLSI graph layout problems. *J. Comput. Syst. Sci.* **28**, 300–343 (1984)
5. Chung, F.R.K.: Labeling of graphs. In: Wilson, R.J., Beineke, L.W. (eds.) *Selected Topics in Graph Theory*, vol. 3, pp. 151–168. Academic Press, London (1988)
6. Corless, R.M., Gonnet, G.H., Hare, D.E.G., Jeffrey, D.J., Knuth, D.E.: On the Lambert W function. *Adv. Comput. Math.* **5**, 329–359 (1996)
7. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: *Introduction to Algorithms*, 3rd edn. MIT Press, Cambridge (2009)
8. Diaz, J., Petit, J., Serna, M.: A survey on graph layout problems. *J. ACM Comput. Surv. (CSUR)* **34**(3), 313–356 (2002)
9. Fertin, G., Vialette, S.: On the S -labeling problem. In: *Proceedings of the 5th European Conference on Combinatorics, Graph Theory and Applications (EuroComb)*. *Electronic Notes on Discrete Mathematics*, vol. 34, pp. 273–277 (2009)
10. Golub, G.H., Van Loan, C.F.: *Matrix Computations*, 3rd edn. Johns Hopkins University Press, Baltimore, London (1996)
11. Gutin, G., Rafiey, A., Szeider, S., Yeo, A.: The linear arrangement problem parameterized above guaranteed value. *Theor. Comput. Syst.* **41**(3), 521–538 (2007). Erratum-ibid [12]
12. Gutin, G., Rafiey, A., Szeider, S., Yeo, A.: Corrigendum. the linear arrangement problem parameterized above guaranteed value. *Theor. Comput. Syst.* **53**(4), 690–691 (2013)
13. Harper, L.H.: Optimal numberings and isoperimetric problems on graphs. *SIAM J.* **12**(1), 131–135 (1964)
14. Johnson, D.S., Yannakakis, M., Papadimitriou, C.H.: On generating all maximal independent sets. *Inf. Process. Lett.* **27**, 119–123 (1988)
15. Karger, D.R.: A randomized fully polynomial approximation scheme for all terminal network reliability problem. In: *27th ACM Symposium on Theory of Computing (STOC)*, pp. 11–17 (1995)
16. Karp, R.M.: Mapping the genome: some combinatorial problem arising in molecular biology. In: *24th ACM Symposium on Theory of Computing (STOC)*, pp. 278–285 (1993)
17. Kitaev, S.: *Iterative Methods for Sparse Linear Systems*, 2nd edn. Society for Industrial and Applied Mathematics, Philadelphia (2003)
18. Leighton, F.T.: *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*. Morgan Kaufmann, San Mateo (1993)
19. Lipton, R.J., Tarjan, R.E.: A separator theorem for planar graphs. *SIAM J. Appl. Math.* **36**(2), 177–189 (1979)
20. Mahajan, M., Raman, V.: Parameterizing above guaranteed values: MaxSat and MaxCut. *J. Algorithms* **31**(2), 335–354 (1999)
21. Niedermeier, R.: *Invitation to Fixed Parameter Algorithms*. *Lecture Series in Mathematics and Its Applications*. Oxford University Press, Oxford (2006)
22. Vialette, S.: Packing of $(0, 1)$ -matrices. *Theor. Inform. Appl. RAIRO* **40**(4), 519–536 (2006)
23. Wilf, H.S.: On the number of maximal independent sets in a tree. *SIAM J. Algebraic Discrete Methods* **7**(1), 125–130 (1986)