On the maximal number of leaves in induced subtrees of series-parallel graphs

Moussa Abdenbi
Laboratoire de combinatoire et d'informatique mathématique
Université du Québec à Montréal
abdenbi.moussa@courrier.uqam.ca

Alexandre Blondin Massé
Laboratoire de combinatoire et d'informatique mathématique
Université du Québec à Montréal
blondin_masse.alexandre@uqam.ca

Alain Goupil
Université du Québec à Trois-Rivières
Alain.Goupil@uqtr.ca

Abstract

Given any simple graph G on n vertices and a positive integer $i \leq n$, an induced subtree of size i of G is called *fully leafed* if it has the maximum number of leaves among all induced subtrees of the same size. It has recently been proved that the problem FLIS of finding fully leafed induced subtrees is NP-hard for general graphs. In this extended abstract, we provide recursive formulas to compute the number of leaves in fully leafed induced subtrees appearing in series-parallel graphs. As a byproduct, the problem FLIS is polynomial for this family of graphs.

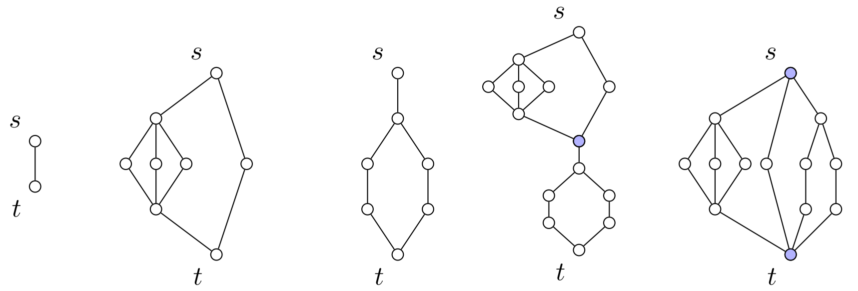
1 Introduction

Given some finite or infinite simple graph G, the study of the subgraphs of G reveals remarkable properties about the whole graph. For instance, it is well known that a graph is planar if and only if it does not contain a subgraph that is a subdivision of the complete bipartite graph $K_{3,3}$ or of the complete graph K_5 [Kur30]. A more general result, due to Robertson and Seymour, states that any family of graphs closed under minors can be equivalently defined by forbidding some finite set of minors [RS04].

In the last 30 years, considerable attention has been devoted to subgraphs that are trees [ESS86, KW91, PTX84, SW05, WAU14]. For instance, Erdős, Saks and Sós considered induced subtrees of maximal size, proving in particular that the problem is NP-hard in general [ESS86]. Around the same time, Payan, Tchuente and Xuong considered trees having many leaves [PTX84] and, in the same spirit, a few years later, Kleitman and West investigated spanning trees with a maximum number of leaves [KW91].

The case of subgraphs that are trees is of particular interest in several fields [BCL05, Yam14, Zak02]. As an example, spanning trees with many leaves turn out to be interesting candidates for minimal energy consumption in wireless newtorks [BCL05]. In chemistry, the Wiener index of chemical trees can be computed from its subtrees [Yam14]. Finally, in the data mining community, it is argued that knowledge can be extracted by inspecting particular subtrees of forests [Zak02].

In: L. Ferrari, M. Vamvakari (eds.): Proceedings of the GASCom 2018 Workshop, Athens, Greece, 18–20 June 2018, published at http://ceur-ws.org



(a) (b) (c) (d) (e)

Figure 1: Series-parallel graphs. The source and sink are labelled by s and t. (a) A basic SP-graph. (b) A SP-graph and (c) another SP-graph. (c) Their series composition and (d) their parallel composition. In (d) and (e), the blue vertices are obtained by "merging" some vertices of the two graphs used in the composition.

We are particularly interested in the problem of enumerating fully leafed induced subtrees, i.e., induced subtrees whose number of leaves is maximal with respect to all induced subtrees of the same size [BCGLNV17]. These combinatorial objects were studied both in the general case [BCGLNV17] and in the particular context of tree-like polyforms and polycubes [BCGS17]. Unfortunately, the problem of finding fully leafed induced subtrees is NP-hard in general [BCGLNV17]. One possible approach to tackle such problems consists of finding a natural parameter that allows fixed parameter tractable algorithms.

We consider the case of series-parallel graphs (SP-graphs), for which the problem becomes polynomial. We believe that the ideas presented below are an important step in designing a parametrized algorithm for solving the problem in general. Additionally, since our algorithm is derived from recursive formulas, it also yield useful insights on the shape of fully leafed induced subtrees in series-parallel graphs.

2 Preliminaries

We briefly recall some definitions from graph theory. See [Die10] for more details. Unless otherwise stated, all graphs considered in this text are undirected.

A graph is an ordered pair $G = (V, E)$, where V is its set of *vertices* and E is its set of *edges*. Given a vertex u, we denote by $N_G(u)$ the set of *neighbors* of u in G, i.e., $N_G(u) = \{v \in V \mid \{u, v\} \in E\}$. When the context is clear, we omit the index G and write simply $N(u)$. The *degree* of u is defined by $\deg(u) = |N(u)|$. The *size* of G, denoted by $|G|$, is the total number $|V|$ of vertices. For $U \subseteq V$, the *subgraph of G induced by U*, denoted by $G[U]$, is the graph $G[U] = (U, E \cap \mathcal{P}_2(U))$, where $\mathcal{P}_2(U)$ is the set of all subsets of size 2 of U.

A graph is called a *tree* if it is both connected and acyclic. A vertex $u \in V$ is called a *leaf* of T if $\deg(u) = 1$. The number of leaves of T is denoted by $|T|_{🍃}$. We say that $G[U]$ is an *induced subtree of G* if the graph induced by U is a tree. *Forests* and *induced subforests* are defined similarly by keeping the "acyclic" property and removing the "connected" one.

A *two-terminal graph* is a quadruple $G = (V, E, s, t)$, where (V, E) is a graph and $s, t \in V$, $s \neq t$, we call s the *source* and t the *sink* of G. A two-terminal graph $G = (V, E, s, t)$ is called a *series-parallel graph* (or simply *SP-graph*) if one of the following three conditions is satisfied:

(i) (Basic SP-graph) $V = \{s, t\}$ and $E = \{\{s, t\}\}$;

(ii) (Series composition) There exist two SP-graphs $G_1 = (V_1, E_1, s_1, t_1)$ and $G_2 = (V_2, E_2, s_2, t_2)$ such that $V = V_1 \cup V_2$, $V_1 \cap V_2 = \{t_1\} = \{s_2\}$, $E = E_1 \cup E_2$, $E_1 \cap E_2 = \emptyset$, $s = s_1$ and $t = t_2$. In this case, we write $G = G_1 \bowtie G_2$.

(iii) (Parallel composition) There exist two SP-graphs $G_1 = (V_1, E_1, s, t)$ and $G_2 = (V_2, E_2, s, t)$, where at least one graph among G_1, G_2 is non basic, such that $V = V_1 \cup V_2$, $V_1 \cap V_2 = \{s, t\}$, $E = E_1 \cup E_2$ and $E_1 \cap E_2 = \emptyset$. In this case, we write $G = G_1 \parallel G_2$.

Examples of SP-graphs are depicted in Figure 1. Series-parallel graphs have been studied since the 19th century. Their enumeration was investigated in 1890 by MacMahon [Mac90] and can be found in OEIS as sequence A000084. A few years later, Riordan and Shannon [RS42] provided a formal proof of the recursive

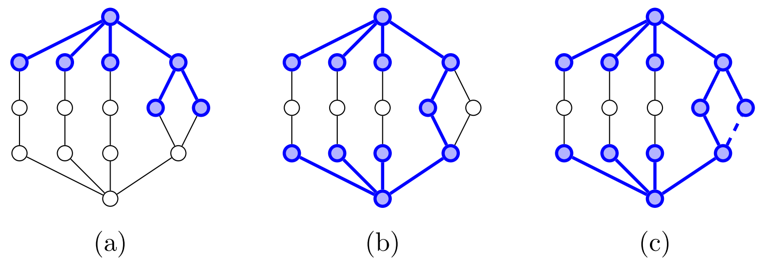


Figure 2: Instances of the leaf induced subtree problem. (a) With $i = 7$ vertices and $\ell = 5$ leaves. (b) With $i = 11$ vertices and $\ell = 6$ leaves. (c) A subtree that is not induced.

i	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$L_G(i)$	2	2	3	4	4	5	5	5	5	6	$-\infty$	$-\infty$	$-\infty$	$-\infty$

Table 1: Leaf function of the SP-graph of Figure 2.

formula given by MacMahon. From there, several papers have been published on their combinatorial properties (see for instance [Fin03] and references therein).

3 Fully Leafed Induced Subtrees

As mentioned in the introduction, we are interested in induced subtrees of series-parallel graphs and in particular those with several leaves. We first recall a definition introduced in [BCGLNV17].

Definition 3.1 (Leaf function, [BCGLNV17]). Given a finite or infinite graph $G = (V, E)$, let $\mathcal{T}_G(i)$ be the family of all induced subtrees of G with exactly i vertices. The *leaf function* of G, denoted by L_G, is the function with domain $\{0, 1, 2, \ldots, |G|\}$ defined by

$$L_G(i) = \max\{|T|_{🍃} : T \in \mathcal{T}_G(i)\},$$

with the convention $\max \emptyset = -\infty$. An induced subtree T of G with i vertices is called *fully leafed* when $|T|_{🍃} = L_G(i)$.

The following problem was considered.

Problem 3.2 (Leaf Induced Subtree Problem, [BCGLNV17]). *Given a simple graph G and two positive integers i and ℓ, does there exist an induced subtree of G with i vertices and ℓ leaves?*

Example 3.3. Let G be the SP-graph illustrated in Figure 2. Positive instances of Problem 3.2 are identified in (a) and (b). It is not hard to prove that both subtrees are fully leafed. More precisely, one can show that $(7, \ell)$ is a negative instance of Problem 3.2 for any $\ell > 5$. Similarly, $(11, \ell)$ is a negative instance of Problem 3.2 for any $\ell > 6$. Notice that the adjective "induced" is important. For example, the blue subgraph illustrated in Figure 2(c) is a subtree, but it is not induced for otherwise the dashed edge would have to be included, therefore creating a cycle. Exhaustive inspection of all induced subtrees show that the leaf function of G is given by Table 1.

4 Main Result

Let G be any SP-graph. We are interested in computing the leaf function L_G of G. An important observation is that induced subtrees of SP-graphs obtained by a *parallel composition* could be obtained by "merging" an *induced subforest* with an induced subtree, i.e., a subtree in a graph and a forest in the other, which is not intuitive. Fortunately, such a subforest factor has a very specific shape: it always has two connected components and it must include both the source and the sink of the SP-graph in which it lives. Hence, we can restrict our attention to those two types of induced subgraphs. Moreover, the following lemma, whose proof is omitted due to lack of space, is key in describing the recursive formulas.

Lemma 4.1. *Let G be a parallel or series composition of two SP-graphs G_1 and G_2. If $T = T_1 \cup T_2$ is a fully leafed induced subtree or subforest of G, where $T_1 \subseteq G_1$ and $T_2 \subseteq G_2$, then T_1 and T_2 are fully leafed induced subtrees or subforests of G_1 and G_2 respectively.*

Hence, by Lemma 4.1, we only need to keep track of the induced subtrees obtained by series or parallel compositions and encode their *status* (leaf or inner vertex and proximity) with respect to the source and the sink.

Definition 4.2. Let $G = (V, E, s, t)$ be an SP-graph and i an integer, with $2 \leq i \leq |G|$. We denote by $L(G, i, \sigma, \tau)$ the maximum number of leaves that can be realized by an induced subtree T of size i of G, such that

$$\sigma = \begin{cases} 0, & \text{if } s \in T, \deg_T(s) > 1; \\ 1, & \text{if } s \in T, \deg_T(s) = 1; \\ 2, & \text{if } s \notin T, |N(s) \cap T| \neq 0; \\ 3, & \text{if } s \notin T, |N(s) \cap T| = 0, \end{cases} \quad \text{and} \quad \tau = \begin{cases} 0, & \text{if } t \in T, \deg_T(t) > 1; \\ 1, & \text{if } t \in T, \deg_T(t) = 1; \\ 2, & \text{if } t \notin T, |N(t) \cap T| \neq 0; \\ 3, & \text{if } t \notin T, |N(t) \cap T| = 0. \end{cases}$$

Similarly, let $F(G, i, \sigma, \tau)$ be the maximum number of leaves that can be realized by an induced subforest of size i whose two connected components T_s and T_t containing s and t respectively are such that

$$\sigma = \begin{cases} 0, & \text{if } \deg_{T_s}(s) > 1; \\ 1, & \text{if } \deg_{T_s}(s) = 1, \end{cases} \quad \text{and} \quad \tau = \begin{cases} 0, & \text{if } \deg_{T_t}(t) > 1; \\ 1, & \text{if } \deg_{T_t}(t) = 1. \end{cases}$$

Clearly, the leaf function L_G of G satisfies

$$L_G(i) = \max_{0 \leq \sigma, \tau \leq 3} L(G, i, \sigma, \tau), \tag{1}$$

for $i = 2, 3, \ldots, |G|$. Therefore, it is sufficient to describe how to compute recursively $L(G, i, \sigma, \tau)$ and $F(G, i, \sigma, \tau)$ to solve Problem 3.2 in the case of series-parallel graphs.

There are two additional useful definitions that we need to introduce. First, let $\text{DistN} = \{2, 3\} \times \{2, 3\} \cup \{0, 1\} \times \{3\} \cup \{3\} \times \{0, 1\}$. Intuitively, when $(\sigma, \tau) \in \text{DistN}$, we are guaranteed that the composition of the induced subtrees/subforests involved will not create a cycle because their respective vertices are "distant". Moreover, for $(a, b) \in \{0, 1\} \times \{0, 1\}$, let ℓ be the *leaf loss function* defined by $\ell(a, b) = a + b$ which indicates the number of leaves that are lost after a composition. For the remainder of this section, we study the different cases, according to the type of composition, in the case of induced subtrees as well as induced subforests.

4.1 Series composition

Assume that $G = G_1 \bowtie G_2$, where $G = (V, E, s, t)$, $G_1 = (V_1, E_1, s_1, t_1)$, $G_2 = (V_2, E_2, s_2, t_2)$. Also, let T be a fully leafed induced subtree of G. There are three cases to consider.

(ST1) *T is included in G_1.* Then we define $ST_1(G, i, \sigma, \tau)$ as the number of leaves of T :

$$ST_1(G, i, \sigma, \tau) = L(G_1, i, \sigma, \tau_1),$$

where

$$\tau = \begin{cases} 2, & \text{if } \{s, t\} \in E_2 \text{ and } \tau_1 \in \{0, 1\}; \\ 3, & \text{otherwise.} \end{cases}$$

(ST2) *T is included in G_2.* Then we define $ST_2(G, i, \sigma, \tau)$ as the number of leaves of T :

$$ST_2(G, i, \sigma, \tau) = L(G_2, i, \sigma_2, \tau)$$

where

$$\sigma = \begin{cases} 2, & \text{if } \{s, t\} \in E_1 \text{ and } \sigma_2 \in \{0, 1\}; \\ 3, & \text{otherwise.} \end{cases}$$

(ST3) *T is included neither in G_1 nor G_2.* Then there exist a fully leafed induced subtree T_1 of G_1 and a fully leafed induced subtree T_2 of G_2 such that $T = T_1 \cup T_2$ and $T_1 \cap T_2 = \{t_1\} = \{s_2\}$. Therefore, the number of leaves of T is

$$ST_3(G, i, \sigma, \tau) = \max_{\substack{(i_1, i_2) \vdash i+1 \\ \tau_1, \sigma_2 \in \{0,1\}}} \{L(G_1, i_1, \sigma, \tau_1) + L(G_2, i_2, \sigma_2, \tau) - \ell(\tau_1, \sigma_2)\}$$

Combining the three cases, we have

$$L(G,i,\sigma,\tau) = \max_{j\in\{1,2,3\}}\{ST_j(G,i,\sigma,\tau)\} \tag{2}$$

Now, assume that F is a fully leafed induced subforest of G whose two connected components are T_s and T_t, with $s \in T_s$ and $t \in T_t$. Once again, we have three cases to consider.

(SF1) *T_s is included in G_1 and T_t is included in G_2.* Then the number of leaves of F is

$$SF_1(G,i,\sigma,\tau) = \max_{\substack{(i_1,i_2)\vdash i\\(\tau_1,\sigma_2)\in\mathrm{DistN}}}\{L(G_1,i_1,\sigma,\tau_1)+L(G_2,i_2,\sigma_2,\tau)\}$$

(SF2) *T_s sticks out of G_1 in G_2 and T_t is included in G_2.* Then

$$SF_2(G,i,\sigma,\tau) = \max_{\substack{(i_1,i_2)\vdash i+1\\\tau_1,\sigma_2\in\{0,1\}}}\{L(G_1,i_1,\sigma,\tau_1)+F(G_2,i_2,\sigma_2,\tau)-\ell(\tau_1,\sigma_2)\}$$

(SF3) *T_s is included in G_1 and T_t sticks out of G_2 in G_1.* In this case, the number of leaves of F is

$$SF_3(G,i,\sigma,\tau) = \max_{\substack{(i_1,i_2)\vdash i+1\\\tau_1,\sigma_2\in\{0,1\}}}\{F(G_1,i_1,\sigma,\tau_1)+L(G_2,i_2,\sigma_2,\tau)-\ell(\tau_1,\sigma_2)\}$$

Combining all three cases, we obtain the expression

$$F(G,i,\sigma,\tau) = \max_{j=1,2,3}\{SF_j(G_1\bowtie G_2,i,\sigma,\tau)\} \tag{3}$$

4.2 Parallel composition

The parallel composition is more intricate, but all cases can be covered using a similar reasoning. Assume that $G = G_1 \parallel G_2$, where $G=(V,E,s,t)$, $G_1=(V_1,E_1,s_1,t_1)$, $G_2=(V_2,E_2,s_2,t_2)$.

As in the last subsection, we first consider the subtree case. Let T be a fully leafed induced subtree of G. We distinguish six cases.

(PT1) *T is included in G_1.* Then the number of leaves of T is

$$PT_1(G,i,\sigma,\tau) = L(G_1,i,\sigma,\tau)$$

(PT2) *T is included in G_2.* In that case, we obtain

$$PT_2(G,i,\sigma,\tau) = L(G_2,i,\sigma,\tau)$$

In the following cases T *lives both in G_1 and G_2.*

(PT3) *$s \in T$ and $t \notin T$.* Then we have $\sigma = 0$, since s is an internal vertex of T. Therefore, the number of leaves in that case is

$$PT_3(G,i,0,\tau) = \max_{\substack{(i_1,i_2)\vdash i+1\\\sigma_1,\sigma_2\in\{0,1\}\\(\tau_1,\tau_2)\in\mathrm{DistN}}}\{L(G_1,i_1,\sigma_1,\tau_1)+L(G_2,i_2,\sigma_2,\tau_2)-\ell(\sigma_1,\sigma_2)\}$$

where $\tau = \min\{\tau_1,\tau_2\}$

(PT4) *$s \notin T$ and $t \in T$.* This case is analogous to (PT3), but now we have $\tau = 0$ and

$$PT_4(G,i,\sigma,0) = \max_{\substack{(i_1,i_2)\vdash i+1\\\tau_1,\tau_2\in\{0,1\}\\(\sigma_1,\sigma_2)\in\mathrm{DistN}}}\{L(G_1,i_1,\sigma_1,\tau_1)+L(G_2,i_2,\sigma_2,\tau_2)-\ell(\tau_1,\tau_2)\}$$

where $\sigma = \min\{\sigma_1,\sigma_2\}$

(PT5) *T can be decomposed in a forest F_1 included in G_1 and a tree T_2 included in G_2.* In this case, observe that $s,t \in F_1, T_2$, which implies $\sigma = 0$ and $\tau = 0$. Therefore, the number of leaves of T is

$$PT_5(G,i,0,0) = \max_{\substack{(i_1,i_2)\vdash i+2\\\sigma_1,\tau_1,\sigma_2,\tau_2\in\{0,1\}}}\{F(G_1,i_1,\sigma_1,\tau_1)+L(G_2,i_2,\sigma_2,\tau_2)$$

$$-\ell(\tau_1,\tau_2)-\ell(\sigma_1,\sigma_2)\}$$

(PT6) *T can be decomposed in a tree T_1 included in G_1 and a forest F_2 included in G_2.* This case is analogous to (PT5) and we obtain

$$PT_6(G,i,0,0) = \max_{\substack{(i_1,i_2)\vdash i+2 \\ \sigma_1,\tau_1,\sigma_2,\tau_2\in\{0,1\}}} \{L(G_1,i_1,\sigma_1,\tau_1)+F(G_2,i_2,\sigma_2,\tau_2)$$

$$-\ell(\tau_1,\tau_2)-\ell(\sigma_1,\sigma_2)\}$$

Combining all six cases, we deduce that

$$L(G,i,\sigma,\tau) = \max_{1\le j\le 6}\{PT_j(G,i,\sigma,\tau)\} \tag{4}$$

Finally, assume that F is a fully leafed induced forest of $G = G_1 \parallel G_2$ and write $F = F_1 \cup F_2$ (F_1 and F_2 possibly empty), where F_1 is included in G_1 and F_2 is included in G_2.

There are nine cases.

(PF1) *$F_2 = \emptyset$.* Then the number of leaves of F is

$$PF_1(G,i,\sigma,\tau) = F(G_1,i,\sigma,\tau)$$

(PF2) *$F_1 = \emptyset$.* Then

$$PF_2(G,i,\sigma,\tau) = F(G_2,i,\sigma,\tau)$$

(PF3) *F_1 is a subforest, F_2 a subtree and $s \in F_2$.* Then

$$PF_3(G,i,0,\tau) = \max_{\substack{(i_1,i_2)\vdash i+1 \\ \sigma_1,\sigma_2\in\{0,1\}}} \{F(G_1,i_1,\sigma_1,\tau)+L(G_2,i_2,\sigma_2,3)-\ell(\sigma_1,\sigma_2)\}$$

(PF4) *F_1 is a subforest, F_2 a subtree and $t \in F_2$.* Then

$$PF_4(G,i,\sigma,0) = \max_{\substack{(i_1,i_2)\vdash i+1 \\ \tau_1,\tau_2\in\{0,1\}}} \{F(G_1,i_1,\sigma,\tau_1)+L(G_2,i_2,3,\tau_2)-\ell(\tau_1,\tau_2)\}$$

(PF5) *F_1 is a subtree, F_2 a subforest and $s \in F_1$.* Then

$$PF_5(G,i,0,\tau) = \max_{\substack{(i_1,i_2)\vdash i+1 \\ \sigma_1,\sigma_2\in\{0,1\}}} \{L(G_1,i_1,\sigma_1,3)+F(G_2,i_2,\sigma_2,\tau)-\ell(\sigma_1,\sigma_2)\}$$

(PF6) *F_1 is a subtree, F_2 a subforest and $t \in F_1$.* Then

$$PF_6(G,i,\sigma,0) = \max_{\substack{(i_1,i_2)\vdash i+1 \\ \tau_1,\tau_2\in\{0,1\}}} \{L(G_1,i_1,3,\tau_1)+F(G_2,i_2,\sigma,\tau_2)-\ell(\tau_1,\tau_2)\}$$

(PF7) *Both F_1 and F_2 are subtrees, $s \in F_1$ and $t \in F_2$.* Then

$$PF_7(G,i,\sigma,\tau) = \max_{\substack{(i_1,i_2)\vdash i \\ \sigma,\tau\in\{0,1\}}} \{L(G_1,i_1,\sigma,3)+L(G_2,i_2,3,\tau)\}$$

(PF8) *Both F_1 and F_2 are subtrees, $s \in F_2$ and $t \in F_1$.* Then

$$PF_8(G,i,\sigma,\tau) = \max_{\substack{(i_1,i_2)\vdash i \\ \sigma,\tau\in\{0,1\}}} \{L(G_1,i_1,3,\tau)+L(G_2,i_2,\sigma,3)\}$$

(PF9) Finally, *Both F_1 and F_2 are subforests.* Then

$$PF_9(G,i,0,0) = \max_{\substack{(i_1,i_2)\vdash i+2 \\ \sigma_1,\tau_1,\sigma_2,\tau_2\in\{0,1\}}} \{F(G_1,i_1,\sigma_1,\tau_1)+F(G_2,i_2,\sigma_2,\tau_2)$$

$$-\ell(\sigma_1,\sigma_2)-\ell(\tau_1,\tau_2)\}$$

To summarize,

$$F(G,i,\sigma,\tau) = \max_{1\le j\le 9}\{PF_j(G,i,\sigma,\tau)\}$$

We have covered all possible cases. It only remains to describe the behavior of the basis cases.

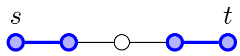


Figure 3: The smallest SP-graph containing a basic forest.

4.3 Basis cases

Let B_T be the basic SP-graph (see Figure 1(a)). Then we have

$$L(B_T, i, \sigma, \tau) = \begin{cases} 2, & \text{if } i = 2 \text{ and } \sigma = \tau = 1; \\ -\infty, & \text{otherwise.} \end{cases} \tag{5}$$

Moreover, for any SP-graph G, we have

$$L(G, 2, 1, \tau) = L(G, 2, \sigma, 1) = 2 \tag{6}$$

for any $\sigma, \tau \in \{0, 1, 2, 3\}$.

Finally, let B_F be the SP-graph of 5 vertices depicted in Figure 3 and F the induced subforest higlighted in blue. Then

$$F(B_F, i, \sigma, \tau) = \begin{cases} 4, & \text{if } i = 4 \text{ and } \sigma = \tau = 1; \\ -\infty & \text{otherwise.} \end{cases} \tag{7}$$

Hence, for any SP-graph $G \neq B_F$ of size at most 5, for all $i \in \{2, 3, \ldots, |G|\}$ and $\sigma, \tau \in \{0, 1\}$, we have

$$F(G, i, \sigma, \tau) = -\infty \tag{8}$$

5 Concluding Remarks

The recursive formulas presented in this paper can be easily translated into a program that generates all fully leafed induced subtrees of a given SP-graph. In the future, we are interested in investigating the enumerative combinatorics of induced subtrees of SP-graphs.

References

[BCGS17] A. Blondin Massé, J. de Carufel, A. Goupil and M. Samson. Fully Leafed Tree-Like Polyominoes and Polycubes. *In: L. Brankovic, J. Ryan, W. F. Smyth (Eds.): Combinatorial Algorithms / 28th International Workshop, IWOCA 2017, Newcastle, NSW, Australia, Lecture Notes in Computer Science*, 10765:206–218, 2017.

[BCGLNV17] A. Blondin Massé, J. de Carufel, A. Goupil, M. Lapointe, E. Nadeau and É Vandomme. Fully leafed induced subtrees. At *https://arxiv.org/abs/1709.09808*, 2018.

[BCL05] A. Boukerche, X. Cheng and J. Linus. A Performance Evaluation of a Novel Energy-Aware Data-Centric Routing Algorithm in Wireless Sensor Networks. *Wireless Networks*, 11(5):619–635, 2005.

[Die10] R. Diestel. *Graph theory.* Graduate Texts in Mathematics, 173, Springer-Verlag Berlin Heidelberg 2014.

[Epp92] D. Eppstein. Parallel recognition of series-parallel graphs. *Information and Computation*, 98(1):41–55, 1992.

[ESS86] P. Erdős, M. Saks and V. T. Sós. Maximum induced trees in graphs. *Journal of Combinatorial Theory. Series B*, 41(1):61–79, 1986.

[Fin03] S. Finch. Series-parallel networks. *Technical report.* Harvard University, 2003.

[KW91] D. J. Kleitman and D. B. West. Spanning trees with many leaves. *SIAM Journal on Discrete Mathematics.*, 4(1):99–106, 1991.

[Kur30] C. Kuratowski. Sur le problème des courbes gauches en Topologie. *Fundamenta Mathematicae*, 15(1):271–283, 1930.

[Mac90] P. A. MacMahon. Yoke-Chains and Multipartite Compositions in connexion with the Analytical forms called "Trees". *Proceedings of the London Mathematical Society*, 1(1):330–346, 1890.

[PTX84] C. Payan, M. Tchuente and N. H. Xuong. Arbres avec un nombre maximum de sommets pendants. *Discrete Mathematics*, 49(3):267–273, 1984.

[RS42] J. Riordan and C. E. Shannon. The Number of Two-Terminal Series-Parallel Networks *Journal of Mathematics and Physics*, 21(1-4):83–93, 1942.

[RS04] N. Robertson and P. D. Seymour. Graph Minors. XX. Wagner's conjecture. *Journal of Combinatorial Theory, Series B*, 92(2):325–357, 2004.

[SW05] L. A. Székely and H. Wang. Arbres avec un nombre maximum de sommets pendants. *Discrete Mathematics*, 34(1):138–155, 2005.

[WAU14] K. Wasa, H. Arimura and T. Uno. Efficient enumeration of induced subtrees in a K-degenerate graph. *In: H.-K- Ahn, C.-S. Shin (Eds.): Algorithms and computation / 25th International Symposium, ISAAC 2014, Jeonju, Korea*, Lecture Notes in Computer Science 8889, Springer, Cham, 2014, pp. 94–102. Springer, Cham, 2014.

[Yam14] M. Yamuna. Wiener index of chemical trees from its subtree. *Der Pharma Chemica*, 6(5):235–242, 2014

[Zak02] M. J. Zaki. Efficiently Mining Frequent Trees in a Forest. *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* KDD '02, pp. 71–80. ACM, 2002.