Filomat 38:3 (2024), 1085–1099
https://doi.org/10.2298/FIL2403085D

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

An inverse result for Wang's theorem on extremal trees

Ivan Damnjanović[a,*], Žarko Ranđelović[b]

[a]*Faculty of Electronic Engineering, University of Niš, Niš, Serbia & Diffine LLC, San Diego, California, USA*
[b]*Centre for Mathematical Sciences, University of Cambridge, Cambridge, UK*

Abstract. It was recently noted by Damnjanović et al. [MATCH Commun. Math. Comput. Chem. **90** (2023), 197–202] that the problem of finding a tree which minimises or maximises the Sombor index among all the trees with a given degree sequence fits within the framework of results by Hua Wang from [Cent. Eur. J. Math. **12** (2014), 1656–1663]. Here, we extend these results by providing an inverse for the aforementioned theorem by Wang. In other words, for any fixed symmetric function f satisfying a monotonicity condition that

$$f(x,a) + f(y,b) > f(y,a) + f(x,b) \quad \text{for any } x > y \text{ and } a > b,$$

we characterise precisely the set of all the trees minimising or maximising the sum $f(\deg x, \deg y)$ over all the adjacent pairs of vertices x and y, among the trees with a given degree sequence.

1. Introduction

Let $\mathcal{T}_D$ denote the set of all trees on n vertices that have a fixed degree sequence $D = (d_1, d_2, \dots, d_n)$. For a symmetric function $f\colon \mathbb{N} \times \mathbb{N} \to \mathbb{R}$, and a tree T in $\mathcal{T}_D$, we write $R_f(T)$ for the sum

$$R_f(T) = \sum_{u \sim v} f(\deg_G(u), \deg_G(v)). \tag{1}$$

Such graph invariants are very natural in their own right, and are studied considerably in chemical graph theory, where they are typically referred to as topological indices and are applied to describe a particular structural property of a given graph of interest (see, for example, [3, 5–9, 11] and the references therein). Some examples can be found in Table 1.

2020 *Mathematics Subject Classification*. Primary 05C35; Secondary 05C05, 05C07, 05C92.
Keywords. tree, degree sequence, adjacent vertices, graph invariant, algorithm, construction, extremal problem.
Received: 25 May 2023; Revised: 14 July 2023; Accepted: 09 August 2023
Communicated by Paola Bonacini
The first author is supported by Diffine LLC.
* Corresponding author: Ivan Damnjanović
Email addresses: ivan.damnjanovic@elfak.ni.ac.rs, ivan@diffine.com (Ivan Damnjanović), zr233@cam.ac.uk (Žarko Ranđelović)

R_f	$f(x,y)$
Randić index	$\frac{1}{\sqrt{xy}}$
first Zagreb index	$x+y$
second Zagreb index	xy
second modified Zagreb index	$\frac{1}{xy}$
geometric–arithmetic index	$\frac{2\sqrt{xy}}{x+y}$
harmonic index	$\frac{2}{x+y}$
sum–connectivity index	$\frac{1}{\sqrt{x+y}}$
atom–bond connectivity index	$\sqrt{\frac{x+y-2}{xy}}$
Sombor index	$\sqrt{x^2+y^2}$

Table 1: Some topological indices R_f together with their corresponding functions $f(x,y)$.

Actually, usually these functions f have a monotonicity property that we shall refer to as 'positive polarity', which says that

$$f(x,a) + f(y,b) \geq f(y,a) + f(x,b) \quad \text{for any } x > y \text{ and } a > b. \tag{2}$$

We will also say that f satisfies 'strict positive polarity' if

$$f(x,a) + f(y,b) > f(y,a) + f(x,b) \quad \text{for any } x > y \text{ and } a > b. \tag{3}$$

We mention in passing that positive polarity often arises because f is the restriction to $\mathbb{N} \times \mathbb{N}$ of a function $g\colon [1,+\infty) \times [1,+\infty) \to \mathbb{R}$ such that the mixed second derivative $\frac{\partial^2 f}{\partial x \, \partial y}(x,y)$ exists and is positive on $(1,\infty) \times (1,\infty)$. For such a g it is easy to check that the restriction f satisfies polarity, and this perhaps explains why polarity is so frequently present.

It was recently noted by Damnjanović et al. [1] that the problem of finding a tree which minimises or maximises the Sombor index among all the trees with a given degree sequence fits within the framework of results by Hua Wang [10]. In an earlier paper, Wang investigated the positive polarity functions and showed that the so-called *greedy tree* must maximize R_f on $\mathcal{T}_D$, while an *alternating greedy tree* necessarily minimizes it, for any such function f [10, Theorem 1.1]. The construction algorithms yielding the two aforementioned types of trees are stated as follows.

Definition 1.1 (Greedy trees, Wang [10, Definition 2.1]). *With given vertex degrees, the greedy tree is achieved through the following "greedy algorithm":*

(i) *Label the vertex with the largest degree as v (the root);*

(ii) *Label the neighbors of v as $v_1, v_2, \ldots$, assign the largest degrees available to them such that* $\deg(v_1) \geq \deg(v_2) \geq \cdots$*;*

(iii) *Label the neighbors of v_1 (except v) as $v_{11}, v_{12}, \ldots$ such that they take all the largest degrees available and that* $\deg(v_{11}) \geq \deg(v_{12}) \geq \cdots$*, then do the same for $v_2, v_3, \ldots$;*

(iv) *Repeat **(iii)** for all the newly labeled vertices, always start with the neighbors of the labeled vertex with the largest degree whose neighbors are not labeled yet.*

Definition 1.2 (Alternating greedy trees, Wang [10, Definition 3.1]). *Given the nonincreasing degree sequence $(d_1, d_2, \dots, d_m)$ of internal vertices, the alternating greedy tree is constructed through the following recursive algorithm:*

(i) *If $m - 1 \leq d_m$, then the alternating greedy tree is simply obtained by a tree rooted at r with d_m children, $d_m - m + 1$ of which are leaves and the rest with degrees $d_1, \dots, d_{m-1}$;*

(ii) *Otherwise, $m - 1 \geq d_m + 1$. We produce a subtree T_1 rooted at r with $d_m - 1$ children with degrees $d_1, \dots, d_{d_m - 1}$;*

(iii) *Consider the alternating greedy tree S with degree sequence $(d_{d_m}, \dots, d_{m-1})$, let v be a leaf with the smallest neighbor degree. Identify the root of T_1 with v.*

Here, we note that the greedy tree is always uniquely determined up to isomorphism, while there might exist more than one alternating greedy tree, given the fact that its construction algorithm is not deterministic. It is also not difficult to observe that the same extremal problem for many other degree-based topological indices can be considered analogously (see, for example, [4], and the references therein).

Our research is primarily motivated by Wang's two algorithms and it is our central goal to extend the said results by providing a way to construct the full solution set to the according extremal problems. Bearing this in mind, we offer the following two non-deterministic tree construction algorithms that yield an extremal $\mathcal{T}_D$ tree for a given non-increasing degree sequence $D = (d_1, d_2, \dots, d_n) \in \mathbb{N}^n$ such that $\mathcal{T}_D \neq \emptyset$.

Algorithm 1.

(i) Add a new vertex, assign its desired degree value to d_1 and assign its availability value to d_1 as well.

(ii) For $j = \overline{2, n}$, repeat the following steps until an output tree is reached.

- **(1)** Add a new vertex u and assign its desired degree and availability values both to d_j.
- **(2)** Let X be the set of all the vertices different from u that have a positive availability.
- **(3)** Choose a vertex v from X so that this vertex has the greatest possible desired degree among all the vertices from X.
- **(4)** Add an edge whose endpoints are the vertices u and v and decrease the availabilities of these two vertices by one.

Algorithm 2.

(i) For each $j = \overline{1, n}$, add some new vertex, assign its desired degree value to d_j and assign its availability value to d_j as well.

(ii) Repeat the following steps until exactly $n - 1$ edges have been added so that an output tree is reached.

- **(1)** Let the set X comprise all the pairs (u, v) of vertices with positive availabilities such that:
  - **(a)** u has the minimum possible desired degree out of all the vertices that have a positive availability;
  - **(b)** u and v do not belong to the same component and the sums of availabilities across the respective components where u and v belong are not both equal to one, unless these are the only two components.
- **(2)** Choose an element of X, i.e. some $(u_0, v_0) \in X$, so that v_0 has the greatest possible desired degree among all the v vertices in the (u, v) pairs of X.
- **(3)** Add an edge whose endpoints are the vertices u_0 and v_0 and decrease the availabilities of these two vertices by one.

It is worth pointing out that, provided $\mathcal{T}_D \neq \emptyset$, Algorithm 1 is clearly well defined. Also, after each **(ii)** iteration from Algorithm 2, the total availabilities of all the components must always yield a valid tree degree sequence, again due to $\mathcal{T}_D \neq \emptyset$. For this reason, it is not difficult to see that the corresponding set X can never be empty. This observation assures us that Algorithm 2 is also well defined. We now present the main result of the given paper in the form of the following theorem.

Theorem 1.3. *For some $n \in \mathbb{N}$, let $D \in \mathbb{N}^n$ be a non-increasing sequence of n integers such that $\mathcal{T}_D \neq \emptyset$ and let $f\colon \mathbb{N} \times \mathbb{N} \to \mathbb{R}$ be a discrete symmetric function. We then have:*

(i) If f is a strict positive polarity function, then a tree $T \in \mathcal{T}_D$ attains the maximum R_f value on $\mathcal{T}_D$ if and only if it is constructible by Algorithm 1 and it attains the minimum R_f value on $\mathcal{T}_D$ if and only if it is constructible by Algorithm 2.

(ii) If f is a positive polarity function, then any tree constructible by Algorithm 1 attains the maximum R_f value on $\mathcal{T}_D$ and any tree constructible by Algorithm 2 attains the minimum R_f value on $\mathcal{T}_D$.

The remainder of the paper will focus on providing a full proof of Theorem 1.3. Its structure will be organized as follows. Section 2 will serve to introduce certain preliminary remarks, as well as some auxiliary construction-related terms for the purpose of making the rest of the proof more concise and easier to follow. Afterwards, Sections 3 and 4 will be used to prove the validity of Algorithms 1 and 2, respectively. Finally, Section 5 will finish the paper by disclosing a brief conclusion regarding all the newly obtained results and will give some examples that elaborate how the given algorithms can be used.

We use standard notation where for a graph G, the order is $|G|$ and $E(G)$ is its set of edges. Also, we will consider all graphs to be undirected, finite and simple. Moreover, we shall implement $\deg_G(u)$ in order to signify the degree of some vertex u from the graph G. Finally, it is worth pointing out that all results are trivial for $n = 1$, so we will always assume that $n \geq 2$.

2. Preliminaries

First of all, it is not difficult to demonstrate that the second claim stated in Theorem 1.3 quickly follows from the first. Let f be an arbitrarily chosen positive polarity function and let $D \in \mathbb{N}^n$ be a non-increasing degree sequence such that $\mathcal{T}_D \neq \emptyset$. For any $x, y, a, b \in \mathbb{N}$ such that $x > y$ and $a > b$, we have

$$(x - y)(a - b) > 0 \quad \implies \quad xa + yb > ya + xb,$$

which means that for any parameter $t \in \mathbb{R}$, $t > 0$, the discrete symmetric function $f_t(x, y) = f(x, y) + txy$ is surely a strict positive polarity function. According to the first statement from Theorem 1.3, we have that any tree $T_0 \in \mathcal{T}_D$ constructible by Algorithm 1 certainly maximizes the R_{f_t} value on $\mathcal{T}_D$, for each $t > 0$. In other words, we get

$$R_{f_t}(T_0) \geq R_{f_t}(T) \tag{4}$$

for any $T \in \mathcal{T}_D$ and $t > 0$. Since both sides of Eq. (4) can be viewed as linear functions in t, we are able to simply plug in $t \to 0^+$ in order to reach

$$R_f(T_0) \geq R_f(T),$$

as desired. An analogous argument can be made regarding the R_f minimizing property of any tree constructible by Algorithm 2. Bearing everything in mind, it becomes evident that in order to complete the proof of Theorem 1.3, it is sufficient to prove just the first disclosed statement. For this reason, we shall deal exclusively with strict positive polarity functions f in the remainder of the paper.

Algorithms 1 and 2 represent two tree construction mechanisms that both involve the simple addition of vertices and edges in some particular order. Throughout both algorithms, each vertex is assigned two property values: the desired degree, which signifies the degree that the vertex should have once the construction is completed, and the availability, which determines how many more edges should be incident to the given vertex in order for its degree to match its desired degree, as needed. We will now define certain construction-related auxiliary terms which we will rely on for the sake of making the proof of Theorem 1.3 easier to follow.

We shall refer to the ordered pair $((v_0, v_1, v_2, \ldots, v_{n-1}), (f_1, f_2, \ldots, f_{n-1}))$ as a *scheme* of some tree T of order $n \in \mathbb{N}$ provided that this tree can be obtained via the following simple construction algorithm:

(1) Add the vertex v_0.

(2) For each integer $j = \overline{1, n-1}$, add the vertex v_j and an edge whose endpoints are v_j and the previously added vertex f_j.

Now, we will use the term *positive availability vertex*, or *PA vertex* for short, to denote a vertex whose availability is greater than zero. If some PA vertex has the greatest desired degree among all the PA vertices, we will then refer to this vertex as a *strong positive availability vertex*, or *SPA vertex* for short. Similarly, if a PA vertex has the smallest desired degree among all the PA vertices, we will then call this vertex a *weak positive availability vertex*, or *WPA vertex* for short.

For a given component, we will use the term *total availability* to refer to the sum of availabilities of all of its vertices and we shall denote the total availability of some component C by $t(C)$. We will consider a *uniform component* to be a component such that all of its PA vertices have the same desired degree. Moreover, we will use $\deg(C)$ to signify the desired degree of any PA vertex from the uniform component C. Furthermore, a uniform component that has the total availability equal to one must necessarily have a single PA vertex, and we will call such a component a *cleaf*. If a component is not uniform, but contains only SPA and WPA vertices, we will then refer to it as a *minimum–maximum mixed component*, or *MMM component* for short. Finally, a component that is neither uniform nor an MMM component shall be called a *forbidden component*.

In the rest of the paper, we will take $D = (d_1, d_2, \ldots, d_n) \in \mathbb{N}^n$ to be an arbitrarily chosen fixed non-increasing sequence of n integers such that $\mathcal{T}_D \neq \varnothing$. Bearing in the mind all the newly introduced terms, it is possible to reformulate Algorithms 1 and 2 in a more concise manner, as demonstrated below.

Algorithm 1.

(i) Add a new vertex and assign its desired degree and availability values to d_1.

(ii) For $j = \overline{2, n}$, repeat the following steps until an output tree is reached.

- **(1)** Add a new vertex u and assign its desired degree and availability values both to d_j.
- **(2)** For an arbitrarily chosen SPA vertex $v \neq u$, add an edge whose endpoints are the vertices u and v and decrease the availabilities of these two vertices by one.

Algorithm 2.

(i) For each $j = \overline{1, n}$, add some new vertex and assign its desired degree and availability values to d_j.

(ii) Repeat the following steps until exactly $n-1$ edges have been added so that an output tree is reached.

- **(1)** Let the set X comprise all the pairs (u, v) of PA vertices from distinct components such that u is a WPA vertex and the total availabilities of the respective components where u and v belong are not both equal to one, unless these are the only two components.
- **(2)** Choose an element of X, i.e. some $(u_0, v_0) \in X$, so that v_0 has the greatest possible desired degree among all the v vertices in the (u, v) pairs of X.
- **(3)** Add an edge whose endpoints are the vertices u_0 and v_0 and decrease the availabilities of these two vertices by one.

3. Validity of Algorithm 1

In this section, we will consider an arbitrary strict positive polarity function f and prove that each tree maximizing R_f on $\mathcal{T}_D$ must be constructible by Algorithm 1. Afterwards, we will swiftly demonstrate the converse as well — that each tree constructible by Algorithm 1 surely attains the maximimum R_f value on $\mathcal{T}_D$.

To begin, we point out that each tree surely has at least one scheme (see, for example, [2, Corollary 1.5.2]). However, it becomes convenient to notice that the trees that attain the maximum R_f value on $\mathcal{T}_D$ always possess very specific schemes. Our immediate goal shall be to elaborate on this fact and provide a result that will later be used while proving the extremal property of Algorithm 1. We start with the following auxiliary lemma regarding the degrees of vertices that lie on an arbitrary path.

Lemma 3.1. *Let $T \in \mathcal{T}_D$ be a tree that attains the maximum R_f value on $\mathcal{T}_D$ and let u and v be two of its arbitrarily chosen vertices. For any vertex w that lies on the path from u to v, we necessarily have*

$$\deg_T(w) \geq \min(\deg_T(u), \deg_T(v)).$$

Proof. We shall prove the lemma by contradiction. Let P be the (u, v)-path in T and suppose that there does lie a vertex on P whose degree is below $\min(\deg_T(u), \deg_T(v))$. It is straightforward to see that $\deg_T(u), \deg_T(v) \geq 2$ must hold. For this reason, we can construct a non-trivial path Q from v to some leaf t so that this path is entirely disjoint with P, except for the vertex v.

Now, let p_1 denote the first vertex on the path P whose degree is lower than $\min(\deg_T(u), \deg_T(v))$, and let p_0 be the vertex on this path before it. Similarly, let q_1 be the first vertex on Q whose degree is below $\min(\deg_T(u), \deg_T(v))$, and let q_0 be the vertex on this path before it. Taking everything into consideration, we obtain a (p_0, q_1)-path as depicted in Figure 1 that will be of further interest.

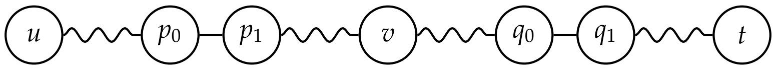

Figure 1: The obtained (p_0, q_1)-path in T, alongside the vertices u and t.

It is clear that the tree T satisfies

$$p_0 \sim p_1, \qquad q_0 \sim q_1, \qquad p_0 \nsim q_0, \qquad p_1 \nsim q_1.$$

Bearing this in mind, we can remove the edges p_0p_1 and q_0q_1 from T and add the edges p_0q_0 and p_1q_1 in order to obtain another tree T_1 whose vertices have the same degrees as in T. Hence, $T_1 \in \mathcal{T}_D$. Furthermore, $R_f(T)$ and $R_f(T_1)$ will have the same summands in Eq. (1) except for those that correspond to the deleted and newly added edges. This immediately implies

$$\begin{aligned} R_f(T_1) - R_f(T) = f(\deg_T(p_0), \deg_T(q_0)) + f(\deg_T(p_1), \deg_T(q_1)) \\ - f(\deg_T(p_0), \deg_T(p_1)) - f(\deg_T(q_0), \deg_T(q_1)). \end{aligned} \tag{5}$$

However, we know that

$$\deg_T(p_0), \deg_T(q_0) \geq \min(\deg_T(u), \deg_T(v)), \qquad \deg_T(p_1), \deg_T(q_1) < \min(\deg_T(u), \deg_T(v)),$$

which swifty leads us to

$$f(\deg_T(p_0), \deg_T(q_0)) + f(\deg_T(q_1), \deg_T(p_1)) > f(\deg_T(q_1), \deg_T(q_0)) + f(\deg_T(p_0), \deg_T(p_1))$$

by virtue of Eq. (3). Now, Eq. (5) tells us that $R_f(T_1) - R_f(T) > 0$ must hold, which is impossible since the tree T attains the maximum R_f value on $\mathcal{T}_D$. Hence, we obtain a contradiction. $\square$

Now, by taking into consideration Lemma 3.1, we are able to formulate and prove the next lemma regarding the constructibility of trees that attain the maximum R_f value.

Lemma 3.2. *If $T \in \mathcal{T}_D$ is a tree that attains the maximum R_f value on $\mathcal{T}_D$, then this tree surely has a scheme $((v_0, v_1, v_2, \ldots, v_{n-1}), (f_1, f_2, \ldots, f_{n-1}))$ such that*

- *for each $j = \overline{0, n-1}$, we have $\deg_T(v_j) = d_{j+1}$;*
- *for all the $1 \le j < h \le n-1$ such that $\deg_T(v_j) = \deg_T(v_h)$, the condition $\deg_T(f_j) \ge \deg_T(f_h)$ must hold.*

Proof. Lemma 3.1 tells us that, for each $j = \overline{\min D, \max D}$, the subgraph of T induced by the set of vertices whose degree is at least j must be a tree. From here, we quickly conclude that we can construct T by simply constructing its subtree induced by the vertices of degree $\max D$, then extending this subtree to the subtree induced by the vertices of degree at least $\max D - 1$, and so on, until we obtain T itself. Thus, the tree T necessarily possesses a scheme $C' = ((v'_0, v'_1, v'_2, \ldots, v'_{n-1}), (f'_1, f'_2, \ldots, f'_{n-1}))$ such that the degrees of the vertices $v'_0, v'_1, v'_2, \ldots, v'_{n-1}$ appear in non-increasing order. This promptly implies $\deg_T(v'_j) = d_{j+1}$ for each $j = \overline{0, n-1}$.

We have obtained a scheme C' that satisfies the first condition given in the lemma. In order to finalize the proof, we will explain how this scheme can be modified so that the second condition surely holds as well. First of all, it is easy to check that the second condition necessarily holds for the vertices of degree $\max D$, hence it becomes sufficient to show that, for any β, $\min D \le \beta < \max D$, the addition of vertices of degree β within the scheme C' can be permuted in some manner so that the second condition becomes satisfied.

The key observation to make is that while T is constructed via the algorithm dictated by C', each vertex of degree β is surely connected to a vertex of degree at least β upon being added. Moreover, each vertex of degree β that is connected to a vertex of degree greater than β can certainly freely be reordered among all the vertices of degree β. In other words, this vertex can be added before or after any other vertex of degree β, given the fact that its initial neighbor is definitely present to begin with. This directly means that we can reorder the addition of all the vertices of degree β so that we first add those whose initial neighbor has the greatest possible degree, then those whose initial neighbor has the second greatest degree, and so on, until we add the vertices of degree β whose initial neighbor also has the degree β, and which cannot freely be reordered. By applying the said transformation on C' for each possible β, $\min D \le \beta < \max D$, we obtain a scheme C that truly satisfies both criteria given in the lemma, which completes the proof. □

By implementing Lemma 3.2, we can immediately prove one half of the desired extremal property of Algorithm 1. This result is disclosed within the following lemma.

Lemma 3.3. *Any tree that attains the maximum R_f value on $\mathcal{T}_D$ must be constructible by Algorithm 1.*

Proof. Let T be any such tree. It is clear that this tree must have a scheme C that satisfies the criteria stated in Lemma 3.2. Now, while T is being constructed via the algorithm dictated by C, suppose that there exists a vertex v such that, upon being added, it is not adjacent to a pre-existing SPA vertex. Let p be such a pre-existing vertex and let q be the vertex that v gets connected to instead. Due to the criteria imposed on C by virtue of Lemma 3.2, we see that none of the vertices of degree $\deg_T(v)$ that are added after v can be adjacent to p either, which means that the vertex p necessarily has a neighbor u in T such that $\deg_T(u) < \deg_T(v)$. Taking everything into consideration, we obtain that the tree T bears a structure as demonstrated in Figure 2.

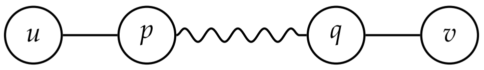

Figure 2: The structure of the tree T.

It is obvious that the tree T satisfies

$$u \sim p, \qquad v \sim q, \qquad u \nsim q, \qquad v \nsim p.$$

If we remove the edges up and vq from T and add the edges uq and vp, we get another tree T_1 whose vertices have the same degrees as in T. For this reason, we have $T_1 \in \mathcal{T}_D$. Using the same logic as in the proof of Lemma 3.1, it is easy to show that

$$\begin{aligned} R_f(T_1) - R_f(T) = f(\deg_T(u), \deg_T(q)) + f(\deg_T(v), \deg_T(p)) \\ - f(\deg_T(u), \deg_T(p)) - f(\deg_T(v), \deg_T(q)). \end{aligned} \tag{6}$$

Taking into consideration that

$$\deg_T(p) > \deg_T(q) \geq \deg_T(v) > \deg_T(u),$$

it becomes straightforward to obtain

$$f(\deg_T(p), \deg_T(v)) + f(\deg_T(q), \deg_T(u)) > f(\deg_T(q), \deg_T(v)) + f(\deg_T(p), \deg_T(u))$$

by directly implementing Eq. (3). Now, by using Eq. (6), this immediately leads us to $R_f(T_1) - R_f(T) > 0$, which is clearly not possible due to the fact that T attains the maximum R_f value on $\mathcal{T}_D$.

Thus, we conclude that while T is being constructed in accordance with the scheme C, the vertices must be added in such a way their degrees yield a non-increasing sequence, with each vertex after the first being connected to a pre-existing SPA vertex. However, this is precisely how Algorithm 1 works, hence it promptly follows that T must indeed be constructible by Algorithm 1. □

We are now finally in position to put all the pieces of the puzzle together and complete the proof of the validity of Algorithm 1.

Proof of the validity of Algorithm 1. If a tree attains the maximum R_f value on $\mathcal{T}_D$, then it is surely constructible by Algorithm 1, by virtue of Lemma 3.3. Thus, in order to finish the validity proof, we need to show that each tree constructible by Algorithm 1 must also attain the maximum R_f value on $\mathcal{T}_D$. Since there are finitely many isomorphism classes among the $\mathcal{T}_D$ trees, there certainly exists a tree $T_0 \in \mathcal{T}_D$ that attains the maximum R_f value on $\mathcal{T}_D$. Due to Lemma 3.3, we know that T_0 is constructible by Algorithm 1. From here we notice that in order to demonstrate that all the trees constructible by Algorithm 1 attain the maximum R_f value on $\mathcal{T}_D$, it is sufficient to prove that they all have the same R_f value.

For each $1 \leq j \leq n$ and $0 \leq k \leq n-1$, let $Y_{j,k}$ denote the sum of availabilities of all the existing vertices of degree k after j vertices have been added in total while executing Algorithm 1. Let the scheme $((v_0, v_1, v_2, \ldots, v_{n-1}), (f_1, f_2, \ldots, f_{n-1}))$ correspond to an execution of Algorithm 1 which yields the tree $T \in \mathcal{T}_D$. It becomes apparent that while adding vertex v_j, the degrees of v_j and f_j can be determined by using the simple expression

$$\begin{aligned} &\deg_T v_j = d_j, \\ &\deg_T f_j = \max\{k \in \mathbb{N} \colon 0 \leq k \leq n-1,\ Y_{j,k} > 0\}. \end{aligned}$$

Besides that, it is possible to obtain the values $Y_{j+1,k}$ in terms of $Y_{j,k}$ by simply setting $Y_{j+1,k} := Y_{j,k}$ for each $0 \leq k \leq n-1$, then increasing the value of $Y_{j+1,\deg_T v_j}$ by $\deg_T v_j - 1$ and then decreasing the value of $Y_{j+1,\deg_T f_j}$ by one. Here, it is important to notice that regardless of how the algorithm is executed, the elements $Y_{j,k}$ depend solely on the given degree sequence D, and not the concrete execution itself. For this reason, the degrees of $v_0, v_1, v_2, \ldots, v_{n-1}$ and $f_1, f_2, \ldots, f_{n-1}$ must be the same in all the executions of Algorithm 1. Given the fact that for any $T \in \mathcal{T}_D$ constructed via the scheme $((v_0, v_1, v_2, \ldots, v_{n-1}), (f_1, f_2, \ldots, f_{n-1}))$, we have

$$R_f(T) = \sum_{j=1}^{n-1} f(\deg_T(v_j), \deg_T(f_j)),$$

it is clear that all the trees constructible by Algorithm 1 must attain the same R_f value, as desired. □

4. Validity of Algorithm 2

In this section, we will consider an arbitrary strict positive polarity function f and prove that each tree minimizing R_f on $\mathcal{T}_D$ must be constructible by Algorithm 2. We will then show that each tree constructible by Algorithm 2 also attains the minimum value of R_f on $\mathcal{T}_D$, thereby completing the proof. We begin by disclosing the following two auxiliary lemmas.

Lemma 4.1. *Let $T \in \mathcal{T}_D$ be a tree that attains the minimum R_f value on $\mathcal{T}_D$. If the tree T contains a path $x_0x_1\cdots x_{n-1}x_n$ of length $n \in \mathbb{N}$, $n \geq 3$ which satisfies $\deg_T(x_0) < \deg_T(x_n)$, then $\deg_T(x_1) \geq \deg_T(x_{n-1})$ must be true.*

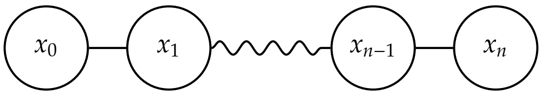

Figure 3: The structure of the path $P = x_0x_1\cdots x_{n-1}x_n$.

Proof. Assume the contrary and let $P = x_0x_1\cdots x_{n-1}x_n$ be a path in T of length $n \in \mathbb{N}$, $n \geq 3$ such that $\deg_T(x_0) < \deg_T(x_n)$ and $\deg_T(x_1) < \deg_T(x_{n-1})$. Now consider the graph T_1 obtained by removing the edges $x_0x_1, x_{n-1}x_n$ and adding the edges x_0x_{n-1} and x_1x_n. Bearing in mind Figure 3, it is evident that T_1 must be a tree. Moreoever, it is straightforward to see that $T_1 \in \mathcal{T}_D$. By implementing Eq. (1), we immediately obtain that

$$\begin{aligned} R_f(T) - R_f(T_1) = {} & f(\deg_T(x_n), \deg_T(x_{n-1})) + f(\deg_T(x_0), \deg_T(x_1)) \\ & - f(\deg_T(x_n), \deg_T(x_1)) - f(\deg_T(x_0), \deg_T(x_{n-1})). \end{aligned}$$

Now, it is sufficient to use Eq. (3) in order to reach $R_f(T) - R_f(T_1) > 0$. Thus, T does not attain the minimum value of R_f on $\mathcal{T}_D$, which is a contradiction. □

Lemma 4.2. *Let $T \in \mathcal{T}_D$ be a tree that attains the minimum R_f value on $\mathcal{T}_D$. Now, let n, b, a be positive integers and suppose that u, v_0, v_1 are vertices in T such that $\deg_T(u) = a$, $\deg_T(v_0) = b$, $\deg_T(v_1) = a$. Furthermore, let $c = \min(a, b)$ and $d = \max(a, b)$. For an arbitrary edge $z_1z_2 \in E(T)$, say that it is good if $\{\deg_T(z_1), \deg_T(z_2)\} = \{c, d\}$. If there is an $i \in \{0, 1\}$ such that there is a path $ux_1\cdots x_nv_i\cdots v_{1-i}y$ in T with $\deg_T(x_n), \deg_T(y) \in [c, d]$, then one of the edges $x_nv_i, v_{1-i}y$ must be good.*

Proof. The proof is trivial to do if $a = b$. We now choose to carry out the proof only for the case when $a < b$, given the fact that the statement can be proved in an entirely analogous manner whenever $b < a$. Thus, we will assume that $c = a$ and $b = d$.

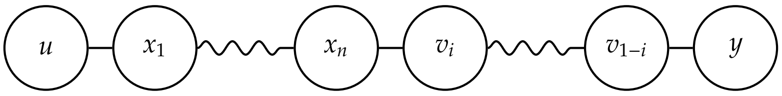

Figure 4: The structure of the path $P = ux_1\cdots x_nv_i\cdots v_{1-i}y$.

Suppose the contrary, that neither x_nv_i nor $v_{1-i}y$ are good edges. Define the paths P, P' to be $P = ux_1...x_nv_i...v_{1-i}y$ and $P' = uv_0...v_1x_1...x_ny$ where we have $V(P) = V(P')$. Let T_1 be the graph obtained by taking T and replacing the edges $ux_1, x_nv_i, v_{1-i}y$ with the edges uv_0, v_1x_1, x_ny. Notice that T_1 is obtained from T when we replace P with P'. This means that T_1 is indeed a tree, and given the fact that P, P' have the same endpoints we have not changed any of the vertex degrees. Hence we obtain that $T_1 \in \mathcal{T}_D$. If $i = 0$, we then have $a < \deg_T(x_n)$ and $\deg_T(y) < b$, which leads us to

$$\begin{aligned} R_f(T) - R_f(T_1) = {} & f(a, \deg_T(x_1)) + f(\deg_T(x_n), b) + f(a, \deg_T(y)) \\ & - f(a, b) - f(a, \deg_T(x_1)) - f(\deg_T(x_n), \deg_T(y)) > 0 \end{aligned}$$

by implementing Eq. (3) together with the aforementioned inequalities. This is a contradiction since T obtains the minimum value of R_f on $\mathcal{T}_D$. The case when $i = 1$ can be resolved in an analogous manner and we choose to leave out the according proof details. $\square$

In the remainder of the section, we will use a, b to denote the desired degrees of the WPA and SPA vertices, respectively. Our next step shall be to use Lemmas 4.1 and 4.2 in order to demonstrate that every $T \in \mathcal{T}_D$ which minimizes R_f is constructible using Algorithm 2. The said result is given in the next lemma.

Lemma 4.3. *Suppose that $T \in \mathcal{T}_D$ minimizes R_f on $\mathcal{T}_D$. Then it is possible to construct T using Algorithm 2 in such a way that at any time after all the vertices have been added and before the tree is fully constructed, the following conditions hold:*

(i) *There is at most one MMM component and all the other components are uniform.*

(ii) *If there is an MMM component, then all the cleaves C have* $\deg(C)$ *equal to a or b.*

(iii) *If there are no MMM components, then each cleaf C satisfying the property* $\deg(C) > a$ *certainly has a degree which is not below the degree of any non-cleaf.*

Proof. Suppose that $k \geq 0$ is the maximum number of edges that we can add using Algorithm 2 such that at every required step the conditions **(i)**, **(ii)** and **(iii)** hold. Note that at the start we have that all components are uniform and all cleaves have the degree one, so **(i)**, **(ii)** and **(iii)** do hold. Let $n = |T|$. If $k = n - 1$, then we are done. Suppose that $k = n - 2$. Then we would have one edge left and we could add it according to Algorithm 2 and there would be a single component remaining so we would be done. Now suppose that $k < n - 2$.

Let $T' \leq T$ be the spanning subgraph of T that we can construct using Algorithm 2 with $k = |E(T')|$. Suppose that T' has l components $C_1, C_2, \ldots, C_l$. Consider the graph T_1 with vertices C_i and where there is an edge C_iC_j if and only if there is an edge in T between C_i and C_j. Note that T_1 is connected and acyclic and hence a tree. Thus, we must have $\deg_{T_1}(C_i) = 1$ for some C_i. Since there are no e, f such that there are two edges in T between C_e and C_f (as that would give a cycle in T), we have that C_i is a cleaf. In particular, there must be a cleaf.

We now choose to split the given problem into two cases.

Case 1. There is no MMM component in T'. We split this case into two further subcases.

Case 1a. Every component of degree a or b is a cleaf. Note that by adding any edge from $T \setminus T'$ we do not obtain any component with zero total availability since we could still add edges to make T. Thus, there is some component that is not a cleaf. Let C be a component that is not a cleaf and has the highest possible degree and let $\deg(C) = \xi$. Also, let u be a PA vertex of desired degree a. There is a shortest path in T from u to C. Let that path be $P = ux_1 \cdots x_m v$ with $m \geq 0$ and $v \in C$. Since C is not a cleaf, there must be some $v_0 \in C$ and a PA vertex $y \in V(T) \setminus C$ such that $y \notin P$ and $v_0 y \in E(T)$. Therefore, $ux_1 \cdots x_m v \cdots v_0 y$ is a path in T.

If $m = 0$, then we may add the edge uv. This is valid in accordance with Algorithm 2 and it is not difficult to realize that all the components would now be uniform. Moreover, the newly formed component containing u, v may or may not be a cleaf, but either way, each cleaf with a degree greater than a would not have a degree lower than any non-cleaf. For this reason, the additional conditions **(i)**, **(ii)** and **(iii)** would all hold as well. This observation would contradict the maximality of k, as desired.

Now, if $m > 0$, then note that $\deg_T(x_1) \leq \xi$ because x_1 must belong to a component that is not a cleaf and not C, since P must enter and then leave that component. This means that $\deg_T(u) = a \leq \deg_T(y)$, as well as $\deg_T(x_1) \leq \xi = \deg_T(v_0)$. This allows us to apply Lemma 4.1 on the path $ux_1 \cdots x_m v \cdots v_0 y$ and deduce that $\deg_T(y) = a$ or $\deg_T(x_1) = \xi$. In any case, there exists an edge whose endpoints are two PA vertices with the desired degrees a and ξ coming from different components, one of which is not a cleaf. Thus, we can add that edge according to Algorithm 2. It is not difficult to establish that all the newly obtained components will be uniform. Also, the newly formed component may or may not be a cleaf, but either way, the condition **(iii)** will hold, as desired. This contradicts the maximality of k once again.

Case 1b. Not all components with degrees a, b are cleaves. As noted earlier, there must exist at least one cleaf. The condition **(iii)** guarantees that there certainly exists a cleaf of degree a or b. Without loss of generality, let there be a cleaf of degree a. We now have that either there is a component of degree b that is not a cleaf, or all the components of degree b are cleaves and then there must be a component of degree a which is not a cleaf. Either way, there are two components C, D of degrees a, b, respectively, such that one of them is a cleaf, while the other is not.

Without loss of generality, let C be a cleaf and let $u \in C$ be the corresponding PA vertex. Following the same argument as in Case 1a, we can show that that Algorithm 2 permits us to add an edge of T whose endpoints have the desired degrees a, b in T. We now have two possibilities — either the newly formed component is uniform or not. If it is uniform, then it must be of degree a or b and it is not difficult to check that all the conditions **(i)**, **(ii)** and **(iii)** must hold. If it is not uniform, this means that we had a non-cleaf of degree b to begin with, which promptly implies that we end up with an MMM component and that all the newly existing cleaves must have the degree a or b. This means that the conditions **(i)**, **(ii)** and **(iii)** all hold. We reach a contradiction regarding the maximality of k.

Case 2. There is an MMM component in T'. Let u be any vertex which is in some cleaf, and by **(ii)**, without loss of generality, let $\deg_T(u) = a$. Let C be the MMM component and let $ux_1 \cdots x_m v_0$ be the shortest path in T from u to C, where $m \geq 0$ and v_0 is a PA vertex. Also, let $v_1 \in C$ be a PA vertex such that $\deg_T(v_0) \neq \deg_T(v_1)$. There must be some $y \notin C$ such that $v_1 y \in E(T)$, which means that $ux_1 \cdots x_m v_0 \cdots v_1 y$ is a path in T.

If $m = 0$, then either $\deg_T(v_0) = b$, in which case we can add the edge uv_0, or $\deg_T(v_1) = b$, and then by Lemma 4.1 we obtain $\deg_T(v_0) = b$ or $\deg_T(y) = a$, thus we can add either the edge uv_0 or $v_1 y$. In each of these scenarios, Algorithm 2 permits us to add an edge in such a way that all the conditions **(i)**, **(ii)** and **(iii)** are satisfies. This can be noticed by using a similar argumentation as done so in Case 1b.

If $m \geq 1$, then by Lemma 4.2, at least one of the two edges $x_m v_0$ or $v_1 y$ will have endpoints with desired degrees a and b and could be added using Algorithm 2. Whatever the case, by adding the said edge, we will connect some component to C and keep at most one MMM component. If there is a new cleaf, it must have degree a or b. Thus, the conditions **(i)**, **(ii)** and **(iii)** will all certainly hold. This once again contradicts the maximality of k. □

We have just shown that all trees that minimize R_f can be constructed using Algorithm 2. Thus, the only thing left to do is to show the converse — that any tree constructed using Algorithm 2 actually minimizes R_f. In order to finalize the desired proof, we will rely on the following lemma which analyzes the behavior of Algorithm 2 while it is being executed.

Lemma 4.4. *Whenever Algorithm 2 is applied on some degree sequence D, at any time after all the vertices have been added and before the tree is fully constructed, the following conditions must hold:*

(i) There are no forbidden components.

(ii) If there is an MMM component, then all the cleaves C have $\deg(C)$ *equal to a or b.*

(iii) If there are no MMM components, then each cleaf C satisfying the property $\deg(C) > a$ *certainly has a degree which is not below the degree of any non-cleaf.*

Proof. We will prove the lemma by induction. Clearly, the lemma statement is true before any edges have been added. Suppose it is true after adding k edges for some $k < n - 2$. By using a similar argument as done so in the proof of Lemma 4.3, there must be a cleaf.

If there is an MMM component, then there must be a cleaf of degree a or b, so Algorithm 2 dictates that an edge should be added whose endpoints have desired degrees a and b. Regardless of which such edge is added, we obtain that all the cleaves must have degrees a and b. Thus, the conditions **(i)**, **(ii)** and **(iii)** must all be satisfied, as desired.

If there is no MMM component, then if all the uniform components of degree a are cleaves, the problem is straightforward to resolve — Algorithm 2 dictates that we should add an edge such that its endpoints have the desired degrees a and ξ, where ξ represents the greatest degree that a non-cleaf component has.

The newly formed component will be uniform and its degree shall be ξ, and from here, it is not difficult to notice that **(i)**, **(ii)** and **(iii)** must all hold.

Finally, if there is no MMM component and there exists at least one component of degree a that is not a cleaf, then Algorithm 2 states that we should add an edge whose endpoints have the desired degrees a and b. However, by doing so, we obtain that the new component is either a uniform component of degree a or b, or an MMM component. If the component is uniform, then it is trivial to see that all the conditions **(i)**, **(ii)** and **(iii)** are satisfied. If the newly generated component is an MMM component, this means that it arose from merging a non-cleaf of degree a and a non-cleaf of degree b. From here, it is evident that all the remaining cleaves must have the degree a or b, which implies that the conditions **(i)**, **(ii)** and **(iii)** all hold once again. $\square$

We are now in the position to implement Lemma 4.4 in order to complete the proof of the validity of Algorithm 2, and thereby finalize the proof of Theorem 1.3.

Proof of the validity of Algorithm 2. If a tree attains the minimum R_f value on $\mathcal{T}_D$, then it must be constructible by Algorithm 2, according to Lemma 4.3. Thus, in order to complete the proof, it is sufficient to demonstrate that each tree constructible by Algorithm 2 must also attain the minimum R_f value on $\mathcal{T}_D$. Similarly to the proof of the validity of Algorithm 1, we note that there are finitely many trees up to isomorphism with a given degree sequence. For this reason, the minimum of R_f must be achieved by some tree and this tree must be constructible by Algorithm 2, due to Lemma 4.3. So, in order to prove that Algorithm 2 only produces trees that minimize R_f, we just need to show that any two trees constructed by Algorithm 2 have the same R_f.

Let T, S be constructed by Algorithm 2 and let $x_1y_1, \dots, x_{n-1}y_{n-1}$ be the edges of T and $z_1t_1, \dots, z_{n-1}t_{n-1}$ be the edges of S in the order in which they were added in Algorithm 2 to construct T, S, respectively. It is enough to show that, for each $k = \overline{1, n-1}$, we have $\{\deg_T(x_k), \deg_T(y_k)\} = \{\deg_S(z_k), \deg_S(t_k)\}$, since this would clearly indicate $R_f(T) = R_f(S)$, by virtue of Eq. (1). Now, for any tree H constructed by Algorithm 2, we will denote $Y_{j,k}(H)$ to be the total sum of availabilities of all vertices of desired degree j after k edges have been added. In order to finalize the proof, it becomes sufficient to show that, for a fixed value $k \in \overline{0, n-1}$, $Y_{j,k}(H)$ is the same for all the trees H and all the values of j. We shall prove this by induction.

Note that the base case for $k = 0$ is true. Suppose that the statement is true up to some k, then notice that $Y_{j,k}(S) = Y_{j,k}(T)$. If $a = b$, then it immediately follows that all the edges to be added throughout the rest of the algorithm will necessarily have endpoints whose desired degrees will all be equal to a. For this reason, it is clear that $Y_{j,k+1}(S) = Y_{j,k+1}(T)$ will hold for each j, and there is nothing left to discuss. Now, suppose that $a < b$. We call an edge *spanning* if its endpoints have degrees a and b. By virtue of Lemma 4.4, there is always a cleaf of degree a or b and, thus, the algorithm allows us to only add spanning edges unless there is no MMM component and all the uniform components of degrees a and b are cleaves.

Now consider what happens after adding the first k edges in both T and S, for some $k < n - 1$. If we suppose that $Y_{j,k+1}(S) = Y_{j,k+1}(T)$ does not hold for each j, we may assume without loss of generality that in T, there is no MMM component and all the uniform components of degrees a or b in T are cleaves. We now point out that for any $a < g < b$, the number of uniform components of degree g must be the same in both T and S as at the start of the algorithm after the vertices have been added but the edges have not. To verify this, we observe that Lemma 4.4 dictates that such uniform components necessarily stay uniform up until k edges have been added. Moreover, the only way for such a component to disappear is if it represents a cleaf which is then merged into another uniform component whose degree is not greater than a. However, this is not possible, since in this scenario, the other component would necessarily not be a cleaf, hence it could be merged with one of the components which have a PA vertex whose desired degree is at least b. Thus, the said edge addition would not be in accordance with Algorithm 2, which is not possible.

Thus, after k edges have been added, both S and T need to have the same number of uniform components whose degree is g, where $a < g < b$. Besides that, the total number of components must be $n - k$ in both trees. Thus, the total number of components containing only PA vertices of desired degrees a and b is the same for T and S. It is not difficult to notice that this can only happen if all components containing PA vertices with desired degrees a and b are cleaves in both T and S. Now, we can see that neither $x_{k+1}y_{k+1}$ nor $z_{k+1}t_{k+1}$

are spanning. Without loss of generality, let $\deg_T(x_{k+1}) = \deg_S(z_{k+1}) = a$. We further have that $\deg_T(y_k)$ is equal to the smallest j, $a \le j < b$ such that $Y_{j,k}(T)$ is strictly larger than the number of uniform components of degree j. However, this value is the same for S and T, hence $\deg_T(y_k) = \deg_S(t_k)$, which completes the proof. □

5. Conclusion

Theorem 1.3 offers a complete solution set for both the R_f maximization and R_f minimization problem on $\mathcal{T}_D$ whenever the discrete symmetric function f is a strict positive polarity. This result makes a substantial contribution to the field of chemical graph theory due to the sheer fact that many adjacent vertex degree based topological indices are yielded by such functions f. For example, by analyzing the topological indices displayed in Table 1, it is straightforward to deduce that a tree $T \in \mathcal{T}_D$ is constructible by Algorithm 1 (Algorithm 2) if and only if it maximizes (minimizes) the Randić index, second Zagreb index, second modified Zagreb index, harmonic index and sum–connectivity index, and if and only if it minimizes (maximizes) the Sombor index.

We finish the paper by comparing Algorithms 1 and 2 with their counterparts disclosed by Wang. While doing so, we will also give two brief examples of how Theorem 1.3 can be used on a concrete valid degree sequence in order to yield the complete solution for the R_f maximization and R_f minimization problem in the case that f is a strict positive polarity function. First of all, it can be said that Algorithm 1 is quite similar to the algorithm given in Definition 1.1 and merely represents a relaxation of its rule set — the incoming vertices do not have to be connected to a fixed available vertex of greatest possible degree, but to any such available vertex. However, this small distinction is precisely what guarantees that Algorithm 1 necessarily yields all the solutions to the R_f maximization problem, while the other algorithm might not. For example, let $D_1 = (4, 4, 3, 3, 2, \underbrace{1, \ldots, 1}_{8 \text{ ones}})$. By implementing the greedy tree construction algorithm, we quickly obtain the uniquely determined greedy tree depicted in Figure 5.

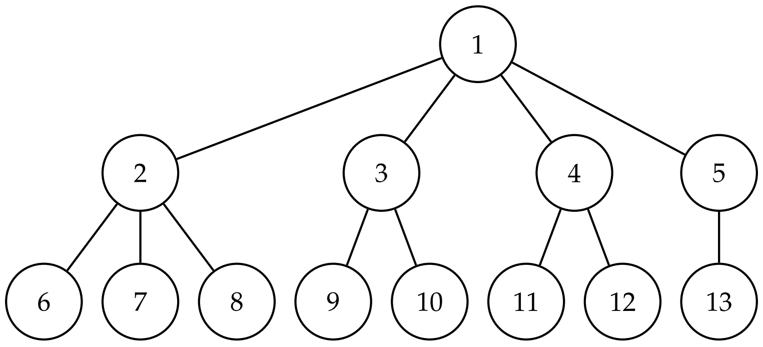

Figure 5: The greedy tree corresponding to the degree sequence $D_1 = (4, 4, 3, 3, 2, 1, 1, 1, 1, 1, 1, 1, 1)$.

The issue with the aforementioned algorithm is that it forces the vertices of degrees three and two to all be adjacent to a fixed vertex of degree four. However, there do exist solutions which do not satisfy this property, which means that the given algorithm fails at generating all the possible solutions. On the other hand, if we apply Theorem 1.3, we are able to swiftly conclude that some tree T attains the maximum R_f value on $\mathcal{T}_{D_1}$ if and only if it belongs to one of the three isomorphism classes shown in Figure 6. This observation is straightforward to notice — the trees constructible by Algorithm 1 are precisely such that the two vertices of degree four are adjacent, while each vertex of degree three or two must have a neighbor of degree four. Thus, there essentially exist exactly three different trees that attain the maximum R_f value on $\mathcal{T}_{D_1}$, with the greedy tree obtained by Wang corresponding to the tree given in Figure 6a. It becomes evident that, although the difference between Algorithm 1 and the greedy tree construction from Definition 1.1 is small, it proves to be crucial if our goal is to obtain all the solutions to the R_f maximization problem.

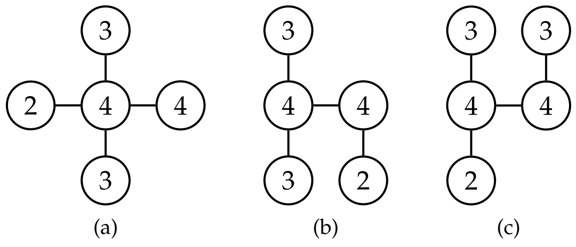

Figure 6: All three isomorphism classes corresponding to the trees constructible by Algorithm 1 for the degree sequence $D_1 = (4, 4, 3, 3, 2, 1, 1, 1, 1, 1, 1, 1, 1)$. Each vertex is labelled by its degree and all the leaves are left out for the sake of brevity.

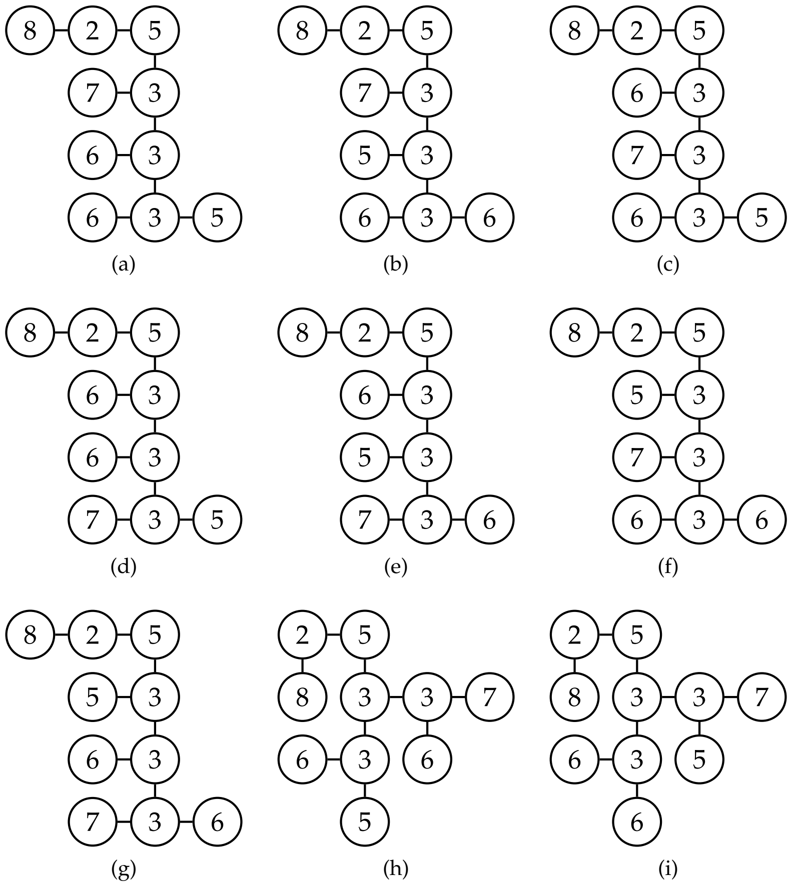

Figure 7: All nine isomorphism classes corresponding to the trees constructible by Algorithm 2 for the degree sequence $D_2 = (8, 7, 6, 6, 5, 5, 3, 3, 3, 2, 1, 1, \ldots, 1)$. Each vertex is labelled by its degree and all the leaves are left out for the sake of brevity.

If we try to compare Algorithm 2 with the alternating greedy tree construction from Definition 1.2, it is not difficult to see that these two algorithms are not that similar. It is convenient to demonstrate this fact through an example. Let $D_2 = (8, 7, 6, 6, 5, 5, 3, 3, 3, 2, \underbrace{1, 1, \ldots, 1}_{30 \text{ ones}})$. If we run Algorithm 2 and use Theorem 1.3, it is possible to obtain nine different isomorphism classes which represent the complete solution set to the R_f minimization problem on $\mathcal{T}_{D_2}$, provided f is a strict positive polarity function. All of these isomorphism classes are depicted in Figure 7. On the other hand, the construction from Definition 1.2 yields only two isomorphism classes [10, Figure 7] that correspond to the trees given in Figures 7e and 7h. The reason why the alternating greedy tree construction gives only two out of nine solutions, while Algorithm 2 generates all nine, is quite easy to explain — the two algorithms use a vastly different rule set and construction methodology. From here, it becomes clear that Theorem 1.3 represents a substantial improvement over the aforementioned earlier construction mechanism.

Acknowledgements

The authors would like to express their gratitude to Imre Leader for all of his supporting comments and remarks.

Conflict of interest

The authors declare that they have no conflict of interest.

References

[1] I. Damnjanović, M. Milošević, D. Stevanović, *A Note on Extremal Sombor Indices of Trees with a Given Degree Sequence*, MATCH Commun. Math. Comput. Chem. **90** (2023), 197–202.
[2] R. Diestel, *Graph Theory*, Springer Berlin, Heidelberg, fifth edition, 2017.
[3] S. Fajtlowicz, *On conjectures of Graffiti–II*, Congr. Numer. **60** (1987), 187–197.
[4] W. Gao, *Trees with Maximum Vertex–Degree–Based Topological Indices*, MATCH Commun. Math. Comput. Chem. **88** (2022), 535–552.
[5] I. Gutman, *Degree-based topological indices*, Croat. Chem. Acta **86(4)** (2013), 351–361.
[6] I. Gutman, *Geometric approach to degree-based topological indices: Sombor indices*, MATCH Commun. Math. Comput. Chem. **86** (2021), 11–16.
[7] I. Gutman, N. Trinajstić, *Graph theory and molecular orbitals. Total φ-electron energy of alternant hydrocarbons*, Chem. Phys. Lett. **17(4)** (1972), 535–538.
[8] A. Miličević, S. Nikolić, N. Trinajstić, *On reformulated Zagreb indices*, Mol. Divers. **8(4)** (2004), 393–399.
[9] M. Randić, *On characterization of molecular branching*, J. Am. Chem. Soc. **97(23)** (1975), 6609–6615.
[10] H. Wang, *Functions on adjacent vertex degrees of trees with given degree sequence*, Cent. Eur. J. Math. **12(11)** (2014), 1656–1663.
[11] B. Zhou, N. Trinajstić, *On a novel connectivity index*, J. Math. Chem. **46** (2009), 1252–1270.