

XIII Latin American Algorithms, Graphs, and Optimization Symposium (LAGOS 2025)

A Weight Function Lemma Heuristic for Graph Pebbling<sup>☆</sup>Guilherme Adamatti Bridi<sup>a,\*</sup>, Franklin de Lima Marquezino<sup>a,\*</sup>, Celina Miraglia Herrera de Figueiredo<sup>a,\*</sup><sup>a</sup>*Federal University of Rio de Janeiro, Rio de Janeiro, Brazil*

---

**Abstract**

Graph pebbling is a problem in which pebbles are distributed across the vertices of a graph and moved according to a specific rule: two pebbles are removed from a vertex to place one on an adjacent vertex. The goal is to determine the minimum number of pebbles required to ensure that any target vertex can be reached, known as the pebbling number. Computing the pebbling number lies beyond NP in the polynomial hierarchy, leading to bounding methods. One of the most prominent techniques for upper bounds is the Weight Function Lemma (WFL), which relies on costly integer linear optimization. To mitigate this cost, an alternative approach is to consider the dual formulation of the problem, which allows solutions to be constructed by hand through the selection of strategies given by subtrees with associated weight functions. To improve the bounds, the weights should be distributed as uniformly as possible among the vertices, balancing their individual contribution. However, despite its simplicity, this approach lacks a formal framework. To fill this gap, we introduce a novel heuristic method that refines the selection of balanced strategies. The method is motivated by our theoretical analysis of the limitations of the dual approach, in which we prove lower bounds on the best bounds achievable. Our theoretical analysis shows that the bottleneck lies in the farthest vertices from the target, forcing surplus weight onto the closer neighborhoods. To minimize surplus weight beyond the theoretical minimum, our proposed heuristic prioritizes weight assignment to the farthest vertices, building the subtrees starting from the shortest paths to them and then filling in the weights for the remaining vertices. Applying our heuristic to Flower snarks and Blanuša snarks, we improve the best-known upper bounds, demonstrating the effectiveness of a structured strategy selection when using WFL.

© 2025 The Authors. Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

Peer-review under responsibility of the Program Committee of LAGOS 2025.

**Keywords:** Graph Pebbling, Weight Function Lemma, Snarks, Heuristics, Graph Theory

---

**1. Introduction**

Graph pebbling is a mathematical game with origins in number theory that has evolved into a broader combinatorial study with applications not only in number theory, but also in optimization, graph theory, probability theory, and computer science [10]. The game can be seen as a transportation problem, where pebbles representing resources to

---

<sup>☆</sup> G.A.B. thanks the financial support of CAPES Grant No. 88887.950125/2024-00. F.L.M. thanks the financial support of CNPq/Universal Grant No. 407296/2021-2. C.M.H.F. thanks the financial support of FAPERJ/CNE Grant No. E-26/204.143/2024, CNPq/Universal Grant No. 404613/2023-3, and CNPq/PQ Grant No. 305356/2021-6.

\* Corresponding author.

*E-mail address:* gabridi@cos.ufrj.br (G.A.B), franklin@cos.ufrj.br (F.L.M), celina@cos.ufrj.br (C.M.H.F)  
1877-0509 © 2025 The Authors. Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

Peer-review under responsibility of the Program Committee of LAGOS 2025.

be delivered to some destination are placed at the vertices of a graph and moved through the edges. Each pebbling move requires spending two pebbles, but only one reaches the adjacent vertex, like a toll paid along the way. The pebbling number of the graph is the minimum number of pebbles needed to guarantee that, no matter how the pebbles are placed on the vertices of the graph, at least one pebble can reach any chosen target vertex—the destination—by some combination of pebbling moves.

The problem of graph pebbling is known to be  $\Pi_2^P$ -complete [11], placing it beyond NP in the polynomial hierarchy—thus making the exact computation of the pebbling number infeasible for large graphs. Consequently, various techniques have been developed to delimit the pebbling number through bounds. One of them is the Weight Function Lemma (WFL), introduced by Hurlbert [9], which provides a useful method for deriving upper bounds. As discussed in Hurlbert [9], applying WFL is not a trivial task, since it involves an integer linear optimization problem whose optimal solution provides a direct upper bound for the pebbling number. Although the associated optimization problem can be solved for small graphs—and since in some cases the relaxation is sufficient, somewhat larger graphs can be handled—the problem becomes very large when scaling the size of the graphs. In general, solving integer linear problems is NP-complete [12].

An alternative approach, explored in Refs. [2, 1, 3], is to work with the dual problem to obtain (at least) near-optimal upper bounds, as it can yield solutions with short certificates in some cases. A key advantage of the dual formulation is the possibility of constructing solutions without relying on optimization techniques, allowing certificates to be obtained by hand, leveraging the combinatorial structure of the graph. Specifically, obtaining a solution to the dual problem involves selecting a set of strategies, where each strategy consists of a subtree rooted at the target vertex and is associated with a weight function that assigns non-negative values to each vertex in the graph. Better bounds are obtained when the sum of the weights of each of the strategies is distributed as evenly as possible among the vertices, balancing the overall contribution required from each vertex. The strength of this combinatorial approach lies in its simplicity. However, it lacks a formal framework to guide its systematic application.

In this sense, we introduce a novel heuristic method to build strategies, motivated by a theoretical analysis of the limitations of the dual formulation of WFL. Our analysis establishes lower bounds on the best achievable upper bounds in this formulation, revealing structural sources of imbalance in the weight distribution over the vertices. These insights guide the design of our heuristic, which we apply to the Blanuša and Flower snarks, improving the upper bounds recently obtained by Adauto et al. [2]. Their work is a pioneering study on the pebbling number of snark graphs, and by focusing on the same graph families, we provide a direct and fair comparison between our heuristic results and the state of the art. Our results highlight the potential of guided strategy construction for improving bounds in challenging graph families.

**Snarks.** In this work, we assume that  $G = (V, E)$  is a simple connected graph. The number of vertices of  $G$  is  $n(G)$ . The distance between vertices  $u$  and  $v$  in a graph  $G$  is denoted by  $d_G(u, v)$ . The eccentricity of a vertex  $v$ , represented by  $e(v)$ , is the maximum distance  $d_G(u, v)$  for any  $u \in V(G)$ . The largest  $e(v)$  of  $G$  is the diameter  $D(G)$ . For  $j \in \mathbb{N}$  and a vertex  $v$ , the  $j^{\text{th}}$  neighborhood of  $v$ , denoted by  $N_j(v)$ , is the set of vertices  $u$  such that  $d_G(u, v) = j$ . In particular, we denote the first neighborhood (or simply the neighborhood) of  $v$  as  $N(v)$ , while the  $e(v)^{\text{th}}$  neighborhood of  $v$ , expressed as  $P(v)$ , is the set of  $v$ -peripheral vertices.

Snarks [5] are a family of graphs that are cubic, bridgeless, 4-edge-chromatic, defined in the context of the Four Color Theorem, which holds if and only if no snark is planar [14]. The Petersen graph, discovered in 1898 [13], is the first and smallest snark, with 10 vertices. Shortly after, in 1946, Danilo Blanuša introduced the Blanuša 1 and 2 snarks, called  $B_1$  and  $B_2$  [5], both with 18 vertices and diameter 4. The present work considers the  $B_2$  snark, shown in Figure 2(a). There are no snarks of order 12, 14, 16, whereas snarks exist for any even order greater than 16. For a detailed history, see Ref. [5]. We also consider the family of Flower snarks  $J_m$  [4], defined for odd  $m = 2k + 1$ ,  $m \geq 3$ . For each  $i \in \pm\{0, 1, \dots, k\}$ , we have vertices  $v_i$ ,  $x_i$ ,  $y_i$ , and  $z_i$ . The vertices  $v_i$ ,  $x_i$ , and  $y_i$  are all adjacent to  $z_i$ , while the vertices  $\{v_i\}$  form the cycle  $C_m$ , with adjacencies given by consecutive indices modulo  $m$ . The vertices  $\{x_i\}$  (resp.  $\{y_i\}$ ) form a path given by the cycle with the edge  $x_k x_{-k}$  (resp.  $y_k y_{-k}$ ) removed. Finally, we add the edges  $x_k y_{-k}$  and  $y_k x_{-k}$  so that the two paths now form one cycle  $C_{2m}$ . We have  $n(J_m) = 4m$  and  $D(J_m) = k + 2$ .

**Graph pebbling.** A configuration  $C$  on a graph  $G$  is a function  $C : V(G) \rightarrow \mathbb{N}$  such that  $C(v)$  is the number of pebbles at the vertex  $v \in V(G)$ . The total number of pebbles in the graph,  $\sum_{v \in V(G)} C(v)$ , is denoted by  $|C|$ . A pebbling move consists of removing 2 pebbles from a vertex  $v$  and adding 1 pebble to a vertex  $u \in N(v)$ . From the configuration

$C$ , the goal of the graph pebbling problem is to get a combination of pebbling moves that reaches a target vertex  $r \in V(G)$ , i.e., that places a pebble on  $r$ . If such a combination of moves exists, we say that the configuration  $C$  is  $r$ -solvable. Otherwise,  $C$  is  $r$ -unsolvable. The problem of deciding on the solvability of target vertices is known to be NP-Complete [15, 11]. The pebbling number of the target vertex  $r$ , denoted by  $\pi(G, r)$ , is the minimum number of pebbles such that all configurations with size  $\pi(G, r)$  are  $r$ -solvable. The pebbling number of the graph, represented by  $\pi(G)$ , is the minimum number of pebbles such that all configurations with size  $\pi(G)$  are  $r$ -solvable for all  $r \in V(G)$ , i.e.,  $\pi(G) = \max_{r \in V(G)} \{\pi(G, r)\}$ . For any graph, the pebbling number is delimited by the well-known basic bounds  $\max\{n(G), 2^{D(G)}\} \leq \pi(G) \leq (n(G) - D(G))(2^{D(G)} - 1) + 1$  [8, 6].

WFL is stated in Lemma 1.1. Before we proceed, we need some definitions. Let  $T$  be a subtree of  $G$  rooted in the target vertex  $r \in V(G)$  such that  $n(T) \geq 2$ . The subtree  $T$  is called an  $r$ -strategy when it is associated with a non-negative function  $\omega_T$ , called weight function, such that  $\omega_T(v) = 0$  if  $v \notin V(T)$  or  $v = r$ , and  $\omega_T(u) \geq 2\omega_T(v)$  for any pair of vertices  $u, v \in V(T) - \{r\}$  such that  $u$  is the parent of  $v$  in  $T$ . We denote the sum of the weights on the vertices of the subtree as  $|\omega_T| = \sum_{v \in V(G) - \{r\}} \omega_T(v)$ . The definition of the weight extends to configurations as, for a configuration  $C$ ,  $\omega_T(C) = \sum_{v \in V(G) - \{r\}} \omega_T(v)C(v)$ .

**Lemma 1.1** (Weight Function Lemma [9]). *Let  $T$  be a  $r$ -strategy of the graph  $G$  with associated weight function  $\omega_T$  and let  $C$  be an  $r$ -unsolvable configuration on  $G$ . Then,  $\omega_T(C) \leq |\omega_T|$ .*

The upper bound on the pebbling number achievable by WFL is given by  $\pi(G, r) \leq \zeta_{G,r} + 1$ , where  $\zeta_{G,r}$  is the optimal solution of the integer linear optimization problem, given by the maximization of  $\sum_{v \in V(G) - \{r\}} C(v)$  subject to  $\omega_T(C) \leq |\omega_T| \forall T \in \mathcal{T}_{\text{set}}$ , with  $\mathcal{T}_{\text{set}}$  being the set of all possible  $r$ -strategies of  $G$ . Note that only the optimal solution can generate a valid upper bound on the pebbling number. In contrast, we can get upper bounds from any solution of the associated dual problem, given by the minimization of  $\sum_{T \in \mathcal{T}_{\text{set}}} \alpha_T |\omega_T|$  subject to  $\sum_{T \in \mathcal{T}_{\text{set}}} \alpha_T \omega_T(v) \geq 1 \forall v \in V(G) - \{r\}$ , where  $\alpha_T \geq 0$  for all  $T \in \mathcal{T}_{\text{set}}$ .

We can manipulate the dual problem as follows. Let  $\mathcal{T} \subseteq \mathcal{T}_{\text{set}}$  be the set of subtrees in which  $\alpha_T > 0$ . Since we can rescale the weight function by a constant factor, let  $1/\alpha = \alpha_T$  for all  $T \in \mathcal{T}$ . Extending the weight function to the set  $\mathcal{T}$ , we have  $\omega_{\mathcal{T}}(v) = \sum_{T \in \mathcal{T}} \omega_T(v)$  and  $|\omega_{\mathcal{T}}| = \sum_{T \in \mathcal{T}} |\omega_T|$ . The dual problem is reduced to minimizing  $|\omega_{\mathcal{T}}|/\alpha$  subject to  $\omega_{\mathcal{T}}(v) \geq \alpha \forall v \in V(G) - \{r\}$ . The optimal value of  $\alpha$  is  $\omega_{\min} = \min_{v \in V(G) - \{r\}} \{\omega_{\mathcal{T}}(v)\}$ —the minimum weight—and we need to minimize  $\lambda_{\mathcal{T}} = |\omega_{\mathcal{T}}|/\omega_{\min}$ . The quantity  $\lambda_{\mathcal{T}}$ , which we call WFL ratio of  $\mathcal{T}$ , provides the upper bound  $\pi(G, r) \leq \lfloor \lambda_{\mathcal{T}} \rfloor + 1$ . The optimal solution of the dual problem  $\lambda_{G,r} = \min_{\mathcal{T} \subseteq \mathcal{T}_{\text{set}}} \{\lambda_{\mathcal{T}}\}$ —the minimum WFL ratio—provides the best dual bound on the pebbling number of the target  $r$ , while  $\lambda_G = \max_{r \in V(G)} \{\lambda_{G,r}\}$  does the same for the graph  $G$ . The lower bounds on the pebbling number result in direct lower bounds on WFL ratio. For instance,  $\pi(G) \geq \max\{n(G), 2^{D(G)}\}$  implies that  $\lambda_G \geq \max\{n(G), 2^{D(G)}\} - 1$ . Figure 1 illustrates the dual approach applied to the Petersen graph.

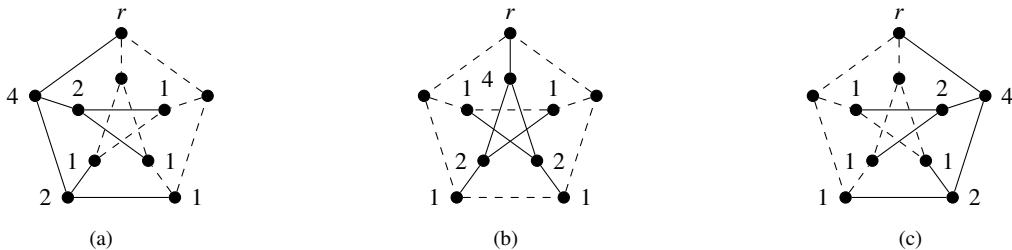


Fig. 1. Adapted representation from Ref. [9] of the optimal  $r$ -strategies highlighted in solid lines (a)  $T_1$ , (b)  $T_2$ , and (c)  $T_3$  for the Petersen graph  $P$ . The Petersen graph is vertex-transitive, allowing us to consider the vertex  $r$  as the only considered target. Note that  $|\omega_{\mathcal{T}}| = 36$ ,  $\omega_{\min} = 4$ , and  $\lambda_{\mathcal{T}} = 9$ . Consequently,  $\pi(P) \leq 10$ . Since  $n(P) = 10$ , it follows that  $\pi(P) = 10$  and  $\lambda_P = 9$ .

## 2. Limits of Weight Function Lemma

Minimizing WFL ratio  $\lambda_{\mathcal{T}}$  involves finding a set of strategies  $\mathcal{T}$ , which we call certificate, that better balances the weight of the function  $\omega_{\mathcal{T}}$  over the vertices, making  $\omega_{\mathcal{T}}$  as uniform as possible. Assuming the most favorable scenario,

i.e.,  $\omega_{\mathcal{T}}(v) = \omega_{\min} \forall v \in V(G) - \{r\}$ , gives the trivial lower bound  $\lambda_{G,r} \geq n(G) - 1$ . When there is a certificate  $\mathcal{T}$  with this property—such as the case of the Petersen graph—we have that  $\pi(G, r) = n(G)$ , a fact known as the Uniform Covering Lemma [9]. Observe that we can write WFL ratio  $\lambda_{\mathcal{T}}$  by adding to  $n(G) - 1$  the weight at each vertex that exceeds  $\omega_{\min}$  and then dividing by  $\omega_{\min}$ . That way,

$$\lambda_{\mathcal{T}} = n(G) - 1 + \frac{\sum_{v \in V(G) - \{r\}} (\omega_{\mathcal{T}}(v) - \omega_{\min})}{\omega_{\min}}. \quad (1)$$

We call  $(\omega_{\mathcal{T}}(v) - \omega_{\min})/\omega_{\min}$  the surplus weight of the vertex  $v$ . To obtain a non-trivial lower bound on the minimum WFL ratio  $\lambda_{G,r}$ , given by Theorem 2.1, we use the relationship between the weight of  $r$ -peripheral vertices and the vertices of the closer neighborhoods. For each  $N_j(r)$ ,  $1 \leq j \leq e(r) - 1$ , in which  $2^{e(r)-j} > |N_j(r)|$ , we get a non-zero lower bound on the surplus weight. We denote by  $I_{\text{sur}}$  the set of such indices  $j$ , i.e.,  $I_{\text{sur}} = \{1 \leq j \leq e(r) - 1 : 2^{e(r)-j} > |N_j(r)|\}$ .

**Theorem 2.1.** *The minimum WFL ratio of a target vertex  $r$  on a graph  $G$  is bounded by*

$$\lambda_{G,r} \geq n(G) - 1 + \sum_{j \in I_{\text{sur}}} (2^{e(r)-j} - |N_j(r)|). \quad (2)$$

*Proof.* Since  $\omega_{\mathcal{T}}(v) \geq \omega_{\min} \forall v \in V(G) - \{r\}$ , we can bound  $\lambda_{G,r}$  from Eq. (1) by  $\lambda_{G,r} \geq n(G) - 1 + \sum_{j=1}^{e(r)-1} \left( \frac{\sum_{v \in N_j(r)} \omega_{\mathcal{T}}(v)}{\omega_{\min}} - |N_j(r)| \right)$ . Let  $u \in P(r)$  and  $1 \leq j \leq e(r) - 1$ . Note that for each subtree  $T \in \mathcal{T}$  in which  $\omega_T(u) > 0$ , there is a vertex  $v_j \in N_j(r)$  in which  $\omega_T(v_j) \geq 2^{e(r)-j} \omega_T(u)$ . Thus,  $\sum_{v \in N_j(r)} \omega_{\mathcal{T}}(v) \geq 2^{e(r)-j} \omega_{\min}$ . On the other hand,  $\sum_{v \in N_j(r)} \omega_{\mathcal{T}}(v) \geq |N_j(r)| \omega_{\min}$ . Combining both bounds with our previous bound on  $\lambda_{G,r}$ , the theorem follows.  $\square$

For each  $j \in I_{\text{sur}}$ , Theorem 2.1 counts the minimum surplus weight in the  $j^{\text{th}}$  neighborhood that originates from the path between the vertices of this neighborhood and the  $r$ -peripheral vertices in the subtrees. From Eq. (2), we see that the minimum surplus weight on  $N_j(r)$  grows exponentially with the distance between the vertices of  $N_j(r)$  and the  $r$ -peripheral vertices, a dependency that matches the exponential nature of the weight function along paths. Furthermore, this growth can be linearly counterbalanced in the number of vertices of  $N_j(r)$ , an intuitive result, since more neighbors imply more paths to distribute and compensate for the weight imbalance. Specifically, if  $|N_j(r)| \geq 2^{e(r)-j}$ , surplus weight is not strictly required in the  $j^{\text{th}}$  neighborhood. If it holds for every  $j$ , i.e.,  $I_{\text{sur}} = \emptyset$ , surplus weight is not strictly required in the whole graph. This is the case of the Petersen graph, with the strategies shown in Figure 1.

To achieve the minimum surplus weight of Theorem 2.1 and thus get the tight bound of Eq. (2), we have the following necessary condition. For any  $j \in I_{\text{sur}}$ ,  $v_j \in N_j(r)$ , and  $u \in P(r)$ , there must be a shortest path from  $r$  to  $u$  passing by  $v_j$ . Equivalently,  $d_{G-\{r\}}(v_j, u) = e(r) - j$  must hold for any  $j, v_j, u$ . To indeed build the tight bound certificate, these shortest paths must be incorporated into the subtrees so that for each vertex  $v_j$ , there must be at least one subtree in which  $v_j$  is part of at least one shortest path from  $r$  to all  $u \in P(r)$ . In addition, we need to ensure that no more surplus weights are added in the subtrees beyond those provided by Theorem 2.1. In particular, the weight of every vertex  $v_j \in N_j(r)$ ,  $j \notin I_{\text{sur}}$ , need to be  $\omega_{\min}$ , while for  $j \in I_{\text{sur}}$ , the sum of the weight in  $N_j(r)$  must be  $2^{e(r)-j} \omega_{\min}$ . An example where we can achieve the tight bound of Eq. (2) is the target vertex  $z_0$  on the Flower graph  $J_3$ , in which the certificate is shown in Figure 3(a). Specifically,  $J_3$  is a graph in which  $I_{\text{sur}} = \{1\}$ , a particular case that plays a determining role in the heuristic method of Section 3. For a more general context, we have the  $d$ -dimensional cube graph  $Q_d$ , whose tight bound certificate can be found in the demonstration of Theorem 16 of Hurlbert [9] paper.

There is an alternative way of expressing the bound of Theorem 2.1. Specifically, let  $I_{\text{no}} = \{1 \leq j \leq e(r) : |N_j(r)| > 2^{e(r)-j}\}$ , the set of indices  $j$  in which we do not establish a non-zero minimum surplus weight for the  $j^{\text{th}}$  neighborhood (excluding the equality case  $2^{e(r)-j} = |N_j(r)|$  and including  $P(r)$ ). Eq. (2) can be written in terms of  $I_{\text{no}}$  as

$$\lambda_{G,r} \geq 2^{e(r)} - 1 + \sum_{j \in I_{\text{no}}} (|N_j(r)| - 2^{e(r)-j}). \quad (3)$$

Note that, combining Eqs. (2) and (3), we can conclude that our lower bound is stronger than the basic bound  $\lambda_G \geq \max\{n(G), 2^{D(G)}\} - 1$ . This alternative expression can be useful to simplify the calculations on graphs with targets in which  $|I_{\text{sur}}| \gg |I_{\text{no}}|$ , such as in graphs with a large diameter. In addition to its practical utility, Eq. (3) also provides a complementary interpretation to Theorem 2.1. Specifically,  $\lambda_{G,r} = 2^{e(r)} - 1$  is achievable when for every  $1 \leq j \leq e(r)$ , the inequality  $\sum_{v \in N_j(r)} \omega_{\mathcal{T}}(v) \geq 2^{e(r)-j} \omega_{\min}$  holds with equality. On the other hand, we also know that  $\sum_{v \in N_j(r)} \omega_{\mathcal{T}}(v) \geq |N_j(r)| \omega_{\min}$ . Thus, if  $|N(r)| \leq 2^{e(r)-j}$ , in principle, there is room to distribute the total weight  $2^{e(r)-j} \omega_{\min}$  on the neighborhood vertices. Otherwise, i.e., if  $|N(r)| > 2^{e(r)-j}$ , the vertices of  $N_j(r)$  force  $\lambda_{G,r}$  to exceed  $2^{e(r)} - 1$  proportional to the difference  $|N_j(r)| - 2^{e(r)-j}$ .

### 3. The heuristic method

The technique used to prove the limitations of WFL on Theorem 2.1 indicates that the bottleneck on the minimization of WFL ratio  $\lambda_{\mathcal{T}}$  is the  $r$ -peripheral vertices. These vertices, while potentially dictating the value of minimum weight  $\omega_{\min}$ , force some vertices in the closer neighborhoods to have surplus weight. Consequently, to get close to the optimal certificate, we need to handle the assignment of weights to the  $r$ -peripheral vertices cautiously. This task involves striking a balance between two goals. The first is to assign as much weight as possible to the  $r$ -peripheral vertices, pushing the value of  $\omega_{\min}$  up. The second is to look for paths that reach the  $r$ -peripheral vertices while minimizing the surplus weight in the closer neighborhoods and, consequently,  $|\omega_{\mathcal{T}}|$ .

Using these insights, we developed a heuristic method to build suitable strategies for a useful application of WFL. Our method is based on the particular case  $I_{\text{sur}} = \{1\}$  of Theorem 2.1. i.e., based on the surplus weight of  $N(r)$ . Broadly speaking, we build one  $r$ -strategy for each neighbor  $v$  of  $r$ . For each  $v$ , we start to build the subtree using shortest paths in the graph  $G - \{r\}$  from  $v$  to all the  $r$ -peripheral vertices, determining the minimum weight. Then, we complete our certificate  $\mathcal{T}$  by adding more vertices to the subtrees to complete the minimum weight in all vertices. Formally, we define the following steps.

1. Denote the neighbors of the target  $r$  as  $v_1, v_2, \dots, v_{|N(r)|}$  and associate each  $v_j$  with a  $r$ -strategy  $T_j$  in such a way that the certificate is  $\mathcal{T} = \{T_1, T_2, \dots, T_{|N(r)|}\}$ . For each  $T_j \in \mathcal{T}$ ,  $v_j$  is the only child of  $r$ .
2. Set the minimum weight as  $\omega_{\min} = \min_{u \in P(r)} \left\{ \sum_{j=1}^{|N(r)|} 2^{e(r)-1-d_{G-\{r\}}(v_j, u)} \right\}$ .
3. For all  $j = 1, 2, \dots, |N(r)|$ , build the so-called trunk of the subtree  $T_j$ . The trunk of  $T_j$  consists of adding shortest paths of the graph  $G - \{r\}$  from  $v_j \in N(r)$  to each  $r$ -peripheral vertex  $u \in P(r)$ . At this step, for each given vertex  $v$  that is added to the subtree  $T_j$ , assign the weight  $\omega_{T_j}(v) = 2^{e(r)-1-d_{T_j}(v_j, v)}$ .
4. For each vertex  $s$  whose summation of the weights on the  $|N(r)|$  trunks built in step 3 has not reached  $\omega_{\min}$ , add  $s$  to the subtrees until  $\omega_{\mathcal{T}}(s) \geq \omega_{\min}$ . We call the branches the edges that we add to the subtrees in step 4. If necessary, increase the weight of the trunk vertices.

The goal of steps 3 and 4 is to minimize the sum of surplus weights in each vertex of the graph by selecting combinations of shortest paths and branches, respectively. For step 3, additional strategies can improve the process in some cases. Let  $P_{\min}(r)$  be the set of minimizers  $\arg \min_{u \in P(r)} \omega_{\min}(u)$  of the equation of the step 2. Note that if  $u \in P_{\min}(r)$ ,  $\omega_{\mathcal{T}}(u) = \omega_{\min}$ . Otherwise,  $\omega_{\mathcal{T}}(u) > \omega_{\min}$ , and the surplus weight on  $u$  gives us two possibilities that can be combined: either simply manually assign less weight to  $u$  and, if possible, decrease the weight of ancestors of  $u$  in the subtree, or replace the shortest path for one of greater length.

Our method can be partially assisted by a computer, automating tasks such as identifying  $r$ -peripheral vertices, computing their distances to neighbors, and listing minimum paths. However, in steps 3 and 4, we go through a decision-making process to choose shortest paths and branches, respectively. Although we have a clear mathematical criterion, which is to minimize surplus weight, we have not proposed an algorithmic process to deal with this task. In the case of step 3, if  $P(r) = P_{\min}(r)$ , we have no surplus weight in the  $r$ -peripheral vertices, and we could solve it by using a brute-force approach to test all combinations of shortest paths. Naturally, the number of combinations is exponential in the worst case. On the other hand, if  $P(r) \neq P_{\min}(r)$ , then we have different approaches to deal with surplus weight on the  $r$ -peripheral vertices, making the decision-making process more complicated. In step 4, we add branches, which is an even less structured process. Thus, our heuristic is feasible for small graphs like Blanuša 2



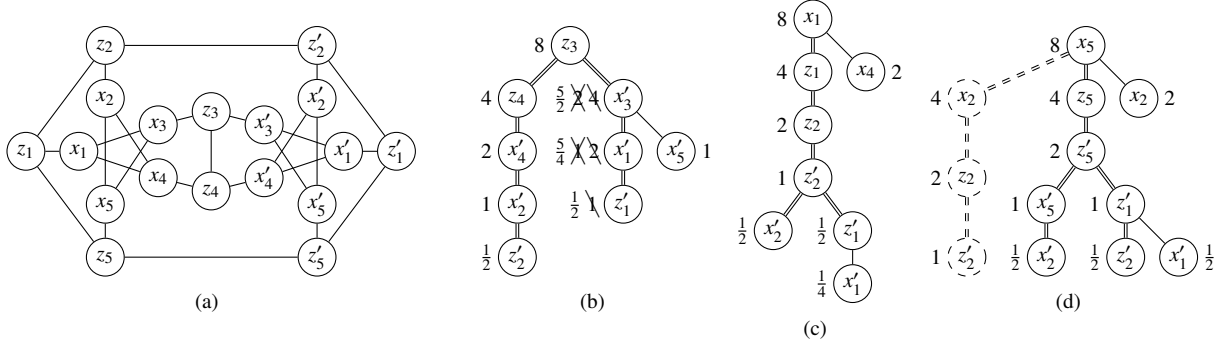


Fig. 2. Application of the heuristic method to the target  $x_3$  of (a) Blanuša 2 graph. We have  $e(x_3) = 4$ ,  $N(x_3) = \{z_3, x_1, x_5\}$ , and  $P(x_3) = \{x'_2, z'_1, z'_2\}$ . The neighbors  $z_3$ ,  $x_1$ , and  $x_5$  of  $x_3$  lead to the strategies (b)  $T_1$ , (c)  $T_2$ , (d)  $T_3$ . Certificate  $\mathcal{T} = \{T_1, T_2, T_3\}$  gives WFL ratio  $\lambda_{\mathcal{T}} = 29.25$ . We omit the root  $x_3$  in the representation of the subtrees. The edges of the trunk are represented by double lines, while the branches are represented by standard lines. The minimum weight, calculated by the equation of the step 2, is  $\omega_{\min} = 2$  and its minimizer is  $P_{\min}(x_3) = \{x'_2\}$ . Therefore, we have surplus weight in  $z'_1$  and  $z'_2$  to work with. For  $z'_2$ , the shortest path from  $x_5$  to  $z'_2$ , indicated by dashed edges and vertices in subtree  $T_3$ , is replaced by the non-shortest path  $x_5, z_5, z'_5, z'_1, z'_2$  in order to reduce the surplus weight. For  $z'_1$ , on the other hand, we take half of the weight of  $z'_1$  in  $T_1$ , as well as of its two closest ancestors, eliminating the surplus weight on the vertex  $x'_3$ . The weights are updated with the original value crossed out with a single strikethrough. Observe that we have no surplus weight beyond the theoretical limit of the algorithm. However, our choice for the trunks of  $\mathcal{T}$  is not the only one with this property. Specifically, in the subtree  $T_1$ , we could reconfigure the subgraph induced by the vertices  $x'_1$  and  $z'_1$  by placing  $x'_1$  as a child of  $x'_4$  and eliminating the vertex  $x'_3$ . For step 4, the vertices that have not yet reached the minimum weight are  $x'_5, x_4, x_2$ , and  $x'_1$ . The vertices  $x'_5, x_4$ , and  $x_2$  are solved directly by adding branches. For  $x'_1$ , which already has weight 1 in the trunk of  $T_1$ , we can reach a total of 1.75 by adding branches in  $T_2$  and  $T_3$ . To reach the minimum of 2 for the vertex  $x'_1$ , we update the weight of  $x'_1$  to  $5/4$  in  $T_1$ , forcing us to create surplus weight in its ancestor  $x'_3$ . In this case, the weights are updated with the original value crossed out with a double strikethrough.

and well-structured infinite families such as Flower graphs, where analyzing smaller cases helps identify structural patterns to construct the general certificate.

*Applications on the snark graphs.* As an initial example, the application of our heuristic approach on the Petersen graph results in the  $r$ -strategies of Figure 1 up to a multiplication factor of 2. We have  $e(r) = 2$  and each of the six vertices of  $P(r)$  is at distance 1 from one neighbor of  $r$  and at distance 2 from the other two neighbors. Therefore, by the equation of the step 2,  $\omega_{\min} = 2$ . In step 3, all the shortest paths from the neighbors of  $r$  to  $r$ -peripheral vertices are unique. Furthermore, since  $P(r) = P_{\min}$ , we have no surplus weight on the  $r$ -peripheral vertices. Step 4 is not required, since all vertices already reach the minimum weight in step 3.

Applying the heuristic to the Blanuša graph  $B_2$  and the Flower graphs  $J_m$  ( $m = 2k + 1, m \geq 3$ ), we improved the best-known upper bounds on  $\lambda_{G,r}$  from Adaauto et al. [2] for most targets. Note that for  $J_3$ , Theorem 2.1 gives  $\lambda_{J_3} \geq 12$ , so the dual approach is not able to improve the bound  $\pi(J_3) \leq 13$  from Adaauto et al. [2]. As a result, we obtained better bounds for the pebbling number of the graphs in Theorems 3.1 ( $B_2$ ) and 3.2 ( $J_m, m \geq 5$ ), where we detail the  $r$ -strategies for a representative target  $r$  in each case.

**Theorem 3.1.** *The pebbling number of the Blanuša graph  $B_2$  is bounded by  $\pi(B_2) \leq 30$ .*

*Proof.* There are six vertex classes, which can be represented, without loss of generality, as the targets  $x_1, x_2, x_3, z_1, z_2, z_3$ . We choose the target  $x_3$  to detail. See Figure 2. The upper bounds on  $\lambda_{G,r}$  for all targets are shown in Table 1. Observe that  $\lambda_{B_2} \leq 29.25$ . Therefore,  $\pi(B_2) \leq 30$  and the theorem follows.  $\square$

**Theorem 3.2.** *For all  $m \geq 5$ , the pebbling number of the Flower graph  $J_m$ , with  $m = 2k + 1$ , is bounded by  $\pi(J_m) \leq \lfloor 2^{k+2}8/5 + 2k - 8/5 \rfloor + 1$ . In particular,  $\pi(J_5) \leq 29$  and  $\pi(J_7) \leq 56$ .*

*Proof.* For any  $m$ ,  $J_m$  has rotational symmetry (with a necessary twist), as well as reflective symmetry that negates subscripts, yielding three vertex classes. In particular, we can choose, without loss of generality,  $x_0, v_0$ , and  $z_0$  as our target vertices. We choose the target  $z_0$  to detail. See Figure 3. The upper bounds on  $\lambda_{G,r}$  for all targets are shown in Table 2. Note that the bottleneck of  $J_m$  ( $m \geq 5$ ) is the vertex  $v_0$ , in which remarkably our heuristic obtained the same result as Adaauto et al. [2]. Thus,  $\pi(J_m) \leq \lfloor 2^{k+2}8/5 + 2k - 8/5 \rfloor + 1$ , as claimed.  $\square$

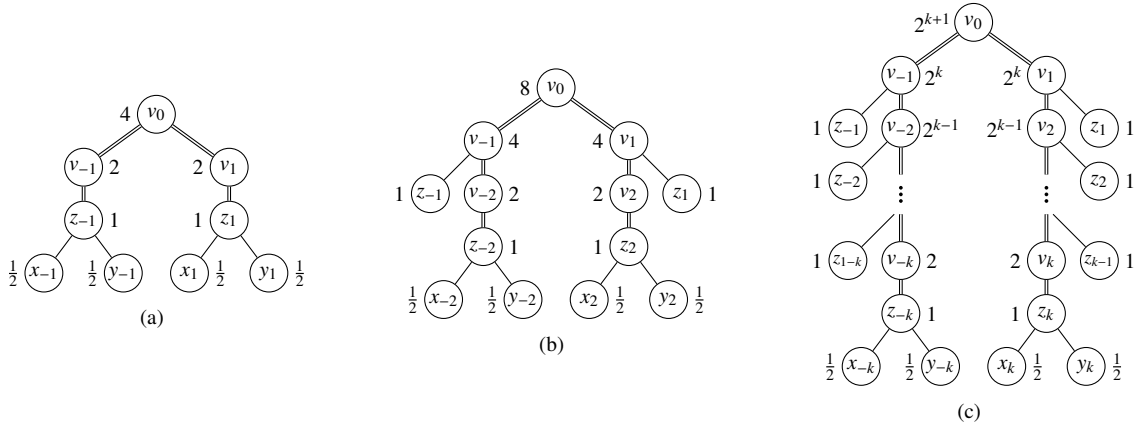


Fig. 3. Application of the heuristic method to the target  $z_0$  of Flower graph  $J_m$ . We start by building the strategies of (a)  $J_3$  and (b)  $J_5$  to arrive at the pattern of the strategies of (c)  $J_m$ , with  $m = 2k + 1, m > 5$ . For each graph, we show only one strategy ( $T_1$ ). The other two strategies are obtained by swapping the vertex labels  $v$  with  $x$  ( $T_2$ ) and  $y$  ( $T_3$ ), respectively. Certificate  $\mathcal{T} = \{T_1, T_2, T_3\}$  gives WFL ratio  $\lambda_{\mathcal{T}} = 2^{k+2}3/2 + 2k - 2$  for  $m \geq 3$ , particularly resulting in a WFL ratio of 12 and 26 for the graphs  $J_3$  and  $J_5$ , respectively. Similarly to the case of Blanuša 2, we omit the root  $z_0$  in the representation of the subtrees, and we represent the trunk edges and branches by double lines and standard lines, respectively. For any  $m \geq 3$ ,  $e(z_0) = k + 2$ ,  $N(z_0) = \{v_0, x_0, y_0\}$ ,  $P(z_0) = \{z_k, z_{-k}\}$ , and the minimum weight, calculated by the equation of the step 2, is  $\omega_{\min} = 3$ . Furthermore,  $P_{\min}(z_0) = P(z_0)$ , and the shortest path in  $G - \{z_0\}$  between each neighbor of  $z_0$  and each  $z_0$ -peripheral vertex is unique. Therefore, each set of trunks is unique to its respective graph. For the branches added in step 4, observe that there is always a missing unit in the weight of the vertices  $v_k, v_{-k}, x_k, x_{-k}, y_k$ , and  $y_{-k}$ . We solve these vertices by adding two branches at each of the vertices  $z_k$  and  $z_{-k}$ , assigning weight  $1/2$  to the new vertices. This is enough to finalize the certificate of the  $J_3$  graph, but observe that vertices  $z_1$  and  $z_{-1}$  have no weight after step 3 in graph  $J_5$ . We solve this case by adding the branches  $v_{-1}z_{-1}$  and  $v_1z_1$ , with weight 1 on both  $z_{-1}$  and  $z_1$ . In general, the vertices  $z_j, z_{-j}$ , for each  $1 \leq j \leq k - 1$ , have no weight in  $J_m$  after step 3 and to them we add the branches  $v_{-j}z_{-j}$  and  $v_jz_j$ , with weight 1 on both  $z_{-j}$  and  $z_j$ , finally getting for the  $z_0$ -strategy shown in (c).

The upper bounds of Adaauto et al. [2] on the pebbling numbers of Blanuša graph  $B_2$  and Flower graphs  $J_m$  ( $m = 2k + 1, m \geq 3$ ) are, respectively, 34 and  $\lfloor 2^{k+2}9/5 + 2k - 18/5 \rfloor + 1$ . Therefore, Theorem 3.1 improves the first in 4 units, while Theorem 3.2 improves the last one by a constant factor of  $9/8$ . Beyond the graph as a whole, Tables 1 (Blanuša graph) and 2 (Flower graphs) make the comparison between our bounds and the bounds of Adaauto et al. [2] with a sharper eye for target vertices, considering the metric of WFL ratio.

Table 1. Comparison between our upper bounds for the minimum WFL ratio of the Blanuša graph  $B_2$  and those previously established by Adaauto et al. [2]. We have improved the upper bounds for almost all roots except  $z_1$ . We highlight targets  $x_2, x_3$ , and  $z_3$ , in which we improve the upper bound on more than 4 units. Observe that Theorem 2.1 gives  $\lambda_{B_2,r} \geq 22$  for all targets. Therefore, the gap between the lower and the upper bound for  $\lambda_{B_2,r}$  lies between 4.6 and 7.25, depending on the target  $r$ .

	WFL ratio upper bounds for Blanuša $B_2$						Graph
	$x_1$	$x_2$	$x_3$	$z_1$	$z_2$	$z_3$	
<b>Our heuristic</b>	29.25	26.6	29.25	26.6	29	27.2	29.25
<b>Adaauto et al. [2]</b>	30	31	$236/7 \approx 33.714$	26.6	30	32	$236/7 \approx 33.714$

*Final remarks.* Our work goes in the opposite, yet complementary, direction of recent optimization-based advances on WFL, such as those by Flocco et al. [7]. Instead, our combinatorial approach explores the graph's structure and is therefore particularly useful for highly structured graphs. In this sense, we have narrowed the known window for the pebbling number of the Blanuša graph  $B_2$  and the Flower graph infinite family  $J_m$  ( $m = 2k + 1, m \geq 5$ ). Our method is grounded in a theoretical lower bound analysis, showing how a deeper understanding of the limitations of the Weight Function Lemma leads to a more structured and effective strategy selection—enhancing both the theoretical insight and the practical applicability of our work. Nonetheless, for the studied graphs, the gap between the lower bounds obtained by Theorem 2.1 and the certificates obtained by the heuristic method—especially for the Flower snarks—is still considerable, indicating potential for improvement in both directions.

Table 2. Comparison between our upper bounds for the minimum WFL ratio of the Flower graphs  $J_m$  ( $m = 2k + 1, m \geq 3$ ) and those previously established by Adaauto et al. [2]. For  $J_3$ , we improve the minimum WFL ratio bound of the target  $z_0$  from 12.8 to 12, which we conclude by Theorem 2.1 that is optimal. For  $m > 7$ , we have improved the  $J_m$  upper bounds for the targets  $x_0$  and  $z_0$  asymptotically by the constant factors of  $17/13$  and  $6/5$ , respectively. For  $v_0$ , we obtained the same result as Adaauto et al. [2], which in particular is the bottleneck for  $\lambda_{J_m}$  in our results. On the other hand, we can conclude using Eq. (3) that Theorem 2.1 gives  $\lambda_{J_m,r} \geq 2^{k+2} + 10$  ( $m \geq 7$ ) for all targets, which implies the gap from our heuristic to our theoretical lower bound lies asymptotically in a constant factor of  $13/10$  and  $8/5$ , depending on the target.

	WFL ratio upper bounds for Flower snarks							
	Our heuristic				Adaauto et al. [2]			
	$J_3$	$J_5$	$J_7$	$J_m (m = 2k + 1, m > 7)$	$J_3$	$J_5$	$J_7$	$J_m (m = 2k + 1, m > 7)$
$x_0$	12.8	27.8	54.6	$2^{k+2}13/10 + 6k - 5$	12.8	29.2	56.8	$2^{k+2}17/10 + 2k - 18/5$
$v_0$	12.8	28	55.6	$2^{k+2}8/5 + 2k - 8/5$	12.8	29.2	55.6	$2^{k+2}8/5 + 2k - 8/5$
$z_0$	12	26	52	$2^{k+2}3/2 + 2k - 2$	12.8	29.2	60	$2^{k+2}9/5 + 2k - 18/5$
<b>Graph</b>	12.8	28	55.6	$2^{k+2}8/5 + 2k - 8/5$	12.8	29.2	60	$2^{k+2}9/5 + 2k - 18/5$

Regarding Theorem 2.1, the tight bound requires that, for each tuple  $(j, v_j, u)$ , where  $j \in I_{\text{sur}}$ ,  $v_j$  is a vertex in the  $j^{\text{th}}$  neighborhood, and  $u$  is a  $r$ -peripheral vertex, there exists a shortest path from  $r$  to  $u$  that passes through  $v_j$ . However, the greater the actual distance between the vertices of the  $j^{\text{th}}$  neighborhood and the  $r$ -peripheral vertices, the harder it becomes to counterbalance the surplus weight caused by the  $r$ -peripheral vertices. Future refinements could take into account, for each neighborhood, a function of the distances between its vertices and the  $r$ -peripheral vertices. For the particular case of the first neighborhood, this function could correspond to the equation of the step 2 of our heuristic.

On the heuristic side, our approach was designed based on the particular case of  $I_{\text{sur}} = \{1\}$  of Theorem 2.1, i.e., by building the subtrees from paths connecting the vertices of the first neighborhood and the  $r$ -peripheral vertices. Future work may explore paths originating from other neighborhoods  $N_j(r)$ , with the choice of  $j$  guided by a criterion that considers the potential surplus weight introduced by the scheme based on each  $N_j(r)$ . In such an approach, one must account not only for the paths from the chosen neighborhood to the  $r$ -peripheral vertices but also for the paths from those vertices back to the root of each subtree. Alternatively, more sophisticated methods could aim to combine paths from multiple neighborhoods simultaneously, seeking a more globally balanced surplus weight distribution. In this context, it becomes increasingly important to automate through some algorithmic procedure the steps 3 and 4 of the heuristic method, for instance by employing greedy criteria, since similar routines are likely to be required in the development of more refined heuristics.

## References

- [1] Adaauto, M., Bardenova, V., da Cruz, M., de Figueiredo, C., Hurlbert, G., Sasaki, D., 2024. Pebbling in Kneser graphs, in: Latin American Symposium on Theoretical Informatics, Lecture Notes in Computer Science, Springer. pp. 46–60.
- [2] Adaauto, M., de Figueiredo, C., Hurlbert, G., Sasaki, D., 2025. On the pebbling numbers of Flower, Blanuša and Watkins snarks. Discrete Applied Mathematics 361, 336–346.
- [3] Alcón, L., Gutierrez, M., Hurlbert, G., 2014. Pebbling in split graphs. SIAM Journal on Discrete Mathematics 28, 1449–1466.
- [4] Campos, C., Dantas, S., de Mello, C., 2011. The total-chromatic number of some families of snarks. Discrete Mathematics 311, 984–988.
- [5] Cavicchioli, A., Meschiari, M., Ruini, B., Spaggiari, F., 1998. A survey on snarks and new results: Products, reducibility and a computer search. Journal of Graph Theory 28, 57–86.
- [6] Chan, M., Godbole, A.P., 2008. Improved pebbling bounds. Discrete Mathematics 308, 2301–2306.
- [7] Flocco, D., Pulaj, J., Yerger, C., 2024. Automating weight function generation in graph pebbling. Discrete Applied Mathematics 347, 155–174.
- [8] Hurlbert, G., 1999. A survey of graph pebbling. Congressus Numerantium 139, 41–64.
- [9] Hurlbert, G., 2017. The weight function lemma for graph pebbling. Journal of Combinatorial Optimization 34, 343–361.
- [10] Hurlbert, G., Kenter, F., 2021. Graph pebbling: A blend of graph theory, number theory, and optimization. Notices of the American Mathematical Society 68, 1900–1913.
- [11] Milans, K., Clark, B., 2006. The complexity of graph pebbling. SIAM Journal on Discrete Mathematics 20, 769–798.
- [12] Papadimitriou, C.H., Steiglitz, K., 1998. Combinatorial Optimization: Algorithms and Complexity. Dover Publications, Nova Iorque.
- [13] Petersen, J., 1898. Sur le théorème de Tait. L'Intermédiaire des Mathématiciens 5, 225–227.
- [14] Tait, P.G., 1880. Remarks on the colouring of maps. Proceedings of the Royal Society of Edinburgh 10, 501–503.
- [15] Watson, N.G., 2005. The complexity of pebbling and cover pebbling. available as arXiv preprint math/0503511 21 Apr 2005 .