

A Graph Pebbling Algorithm on Weighted Graphs

Nándor Sieben

Northern Arizona University, Department of Mathematics and Statistics,
Flagstaff AZ 86011-5717, USA

Abstract

A pebbling move on a weighted graph removes some pebbles at a vertex and adds one pebble at an adjacent vertex. The number of pebbles removed is the weight of the edge connecting the vertices. A vertex is reachable from a pebble distribution if it is possible to move a pebble to that vertex using pebbling moves. The pebbling number of a weighted graph is the smallest number m needed to guarantee that any vertex is reachable from any pebble distribution of m pebbles. Regular pebbling problems on unweighted graphs are special cases when the weight on every edge is 2. A regular pebbling problem often simplifies to a pebbling problem on a simpler weighted graph. We present an algorithm to find the pebbling number of weighted graphs. We use this algorithm together with graph simplifications to find the regular pebbling number of all connected graphs with at most nine vertices.

Submitted: April 2009	Reviewed: August 2009	Revised: December 2009	Accepted: January 2010
	Final: January 2010	Published: February 2010	
Article type: Regular Paper		Communicated by: G. Liotta	

1 Introduction

Graph pebbling has its origin in number theory. It is a model for the transportation of resources. Starting with a pebble distribution on the vertices of a simple connected graph, a *pebbling move* removes two pebbles from a vertex and adds one pebble at an adjacent vertex. We can think of the pebbles as fuel containers. Then the loss of the pebble during a move is the cost of transportation. A vertex is called *reachable* if a pebble can be moved to that vertex using pebbling moves. The *pebbling number* of a graph is the minimum number of pebbles that guarantees that every vertex is reachable. There are many different variations of pebbling. For a comprehensive list of references for the extensive literature see the survey papers [9, 10].

One of our goals is to develop an algorithm that finds the pebbling number in a reasonable amount of computing time. Our approach is similar to the algorithmic approach of [1]. Of course this goal is only realistic for relatively small graphs since finding the pebbling number is a Π_2^P -complete problem [13]. In spite of this difficulty, the implementation of our algorithm is sufficiently fast to run on a large number of graphs. So it can be used for checking conjectures and finding interesting examples. We also believe that our theory views pebbling numbers from a new perspective and so it might become a useful tool in future research.

The main idea of the algorithm is that if we know all the *sufficient* distributions from which a given goal vertex is reachable then we can find the *insufficient* distributions from which the goal vertex is not reachable. An insufficient distribution must be smaller than every sufficient distribution. The pebbling number can be found by finding an insufficient distribution with the most pebbles. The problem is that there are too many sufficient distributions. Luckily it suffices to find the *barely sufficient* distributions from which the goal vertex is no longer reachable after the removal of any pebble.

Our algorithm works even if the cost of moving a pebble from one vertex to another varies between different vertices. To take advantage of this, we develop the basic theory of graph pebbling on weighted graphs used in [7].

The generalization is worth the effort since pebbling on many graphs can be simplified if we replace the graph by a weighted graph with fewer edges. For example a tree can be replaced by a weighted graph containing a single edge. Cut vertices, leaves and ears offer the most fruitful simplifications.

We use these simplifications and our algorithm to calculate the pebbling number of all connected graphs with at most nine vertices. We present the spectrum of pebbling numbers in terms of the number of vertices in the graph.

2 Preliminaries

Let G be a simple connected graph. We use the notation $V(G)$ for the vertex set and $E(G)$ for the edge set. We use the standard notation $vu = uv$ for the edge $\{v, u\} \in E(G)$. A *path* of G is a subgraph isomorphic to the path graph P_n

with $n \geq 1$ vertices. A *weighted graph* G_ω is a graph G with a weight function $\omega : E(G) \rightarrow \mathbf{N}$.

A *pebble function* on G is a function $p : V(G) \rightarrow \mathbf{Z}$ where $p(v)$ is the number of pebbles placed at v . A *pebble distribution* is a nonnegative pebble function. The *size* of a pebble distribution p is the total number of pebbles $\|p\| = \sum_{v \in V(G)} p(v)$. The *support* of the pebble distribution p is the set $\text{supp}(p) = \{v \in V(G) \mid p(v) > 0\}$. We are going to use the notation $p(v_1, \dots, v_n, *) = (a_1, \dots, a_n, q(*))$ to indicate that $p(v_i) = a_i$ for $i \in \{1, \dots, n\}$ and $p(w) = q(w)$ for all $w \in V(G) \setminus \{v_1, \dots, v_n\}$.

If $vu \in E(G)$ then the *pebbling move* $(v \rightarrow u)$ on the weighted graph G_ω removes $\omega(vu)$ pebbles at vertex v and adds one pebble at vertex u , more precisely, it replaces the pebble function p with the pebble function

$$p_{(v \rightarrow u)}(v, u, *) = (p(v) - \omega(vu), p(u) + 1, p(*)).$$

Note that the resulting pebble function $p_{(v \rightarrow u)}$ might not be a pebble distribution even if p is.

The *inverse* of the pebbling move $(v \rightarrow u)$ is denoted by $(v \rightarrow u)^{-1}$. The inverse removes a pebble from u and adds two pebbles at v , that is, it creates the new distribution $p_{(v \rightarrow u)^{-1}}(v, u, *) = (p(v) + 2, p(u) - 1, p(*))$. Note that $(v \rightarrow u)^{-1}$ is not a pebbling move.

A *pebbling sequence* is a finite sequence $s = (s_1, \dots, s_k)$ of pebbling moves. The pebble function gotten from the pebble function p after applying the moves in s is denoted by p_s . The concatenation of the pebbling sequences $r = (r_1, \dots, r_k)$ and $s = (s_1, \dots, s_l)$ is denoted by $rs = (r_1, \dots, r_k, s_1, \dots, s_l)$.

A pebbling sequence (s_1, \dots, s_n) is *executable* from the pebble distribution p if $p_{(s_1, \dots, s_i)}$ is nonnegative for all $i \in \{1, \dots, n\}$. A vertex x of G is *t-reachable* from the pebble distribution p if there is an executable pebbling sequence s such that $p_s(x) \geq t$. We say x is *reachable* if it is 1-reachable.

We write $\pi_t(G_\omega, x)$ for the minimum number m such that x is t -reachable from every pebble distribution of size m . We use the notation $\pi(G_\omega, x)$ for $\pi_1(G_\omega, x)$. The *t-pebbling number* $\pi_t(G_\omega)$ is $\max\{\pi_t(G_\omega, x) \mid x \in V(G)\}$. The *pebbling number* $\pi(G_\omega)$ is the 1-pebbling number $\pi_1(G_\omega)$.

If $\omega(e) = 2$ for all $e \in E(G)$ then $\pi(G_\omega) = \pi(G)$ is the usual unweighted pebbling number. So we allow the weight function ω to be defined only on a subset of $V(G)$ and use the default weight of 2 for edges where ω is undefined.

Changing the order of moves in an executable pebbling sequence s may result in a sequence r that is no longer executable. On the other hand the ordering of the moves has no effect on the resulting pebble function, that is, $p_s = p_r$. This motivates the following definition.

Given a multiset S of pebbling moves on the weighted graph (G_ω) , the *transition digraph* $T(G, S)$ is a directed multigraph whose vertex set is $V(G)$, and each move $(v \rightarrow u)$ in S is represented by a distinct directed edge (v, u) . The transition digraph of a pebbling sequence $s = (s_1, \dots, s_n)$ is $T(G, s) = T(G, S)$, where $S = \{s_1, \dots, s_n\}$ is the multiset of moves in s . Let $d_{T(G, S)}^-$ denote the in-degree and $d_{T(G, S)}^+$ the out-degree in $T(G, S)$. We simply write d^- and d^+

if the transition digraph is clear from context. It is easy to see that the pebble function gotten from p after applying the moves in a multiset S of pebbling moves in any order satisfies

$$p_S(v) = p(v) + d_{T(G,S)}^-(v) - \sum \{\omega(vu) \mid (v,u) \in E(T(G,S))\}$$

for all $v \in G$. For unweighted graphs the formula simplifies to

$$p_S(v) = p(v) + d_{T(G,S)}^-(v) - 2d_{T(G,S)}^+(v).$$

3 Cycles in the transition digraph

In this section we present a version of the No-Cycle Lemma [6, 13, 14]. If the pebbling sequence s is executable from a pebble distribution p then we clearly must have $p_s \geq 0$. We say that a multiset S of pebbling moves is *balanced* with a pebble distribution p at vertex v if $p_S(v) \geq 0$. The multiset S is *balanced* with p if S is balanced with p at all $v \in V(G)$, that is, $p_S \geq 0$. We say that a pebbling sequence s is *balanced* with p if the multiset of moves in s is balanced with p . The balance condition is necessary but not sufficient for a pebbling sequence to be executable. A multiset of pebbling moves or a pebbling sequence is called *acyclic* if the corresponding transition digraph has no directed cycles.

Proposition 3.1 *If S is a multiset of pebbling moves on G_ω then there is an acyclic multiset $R \subseteq S$ such that $p_R \geq p_S$ for each pebble function p on G .*

Proof: Let p be a pebble function on G . Suppose that $T(G,S)$ has a directed cycle C . Let Q be the multiset of pebbling moves corresponding to the arrows of C and $R = S \setminus Q$. Let u_v be the first vertex from v along C . Then $p_R(v) = p_S(v) - 1 + \omega(vu_v) \geq p_S(v)$ for $v \in V(C)$ and $p_R(v) = p_S(v)$ for $v \in V(G) \setminus V(C)$.

We can repeat this process until we eliminate all the cycles. We finish in finitely many steps since every step decreases the number of pebbling moves. \square

Definition 3.2 *Let S be a multiset of pebbling moves on G . An element $(v \rightarrow u) \in S$ is called an initial move of S if v has indegree 0 in $T(G,S)$. A pebbling sequence s is called regular if s_i is an initial move of $S \setminus \{s_1, \dots, s_{i-1}\}$ for all i .*

It is clear that if the multiset S is balanced with a pebble distribution p and s is an initial move of S then s is executable from p .

Proposition 3.3 *If S is an acyclic multiset then there is a regular sequence s of the elements of S . If S is also balanced with the pebble function p then s is executable from p .*

Proof: If S is acyclic then we must have an initial move t of S . Then $S \setminus \{t\}$ is still acyclic. So we can recursively find the elements of $s = (s_1, \dots, s_k)$ by picking an initial move t of S and then replacing S with $S \setminus \{t\}$ at each step.

Now assume that S is balanced with p . Then $S_i = S \setminus \{s_1, \dots, s_{i-1}\}$ is balanced with $p_{(s_1, \dots, s_{i-1})}$ for all i since $(p_{(s_1, \dots, s_{i-1})})_{S_i} = p_S \geq 0$. Hence the initial move s_i of S_i is executable from $p_{(s_1, \dots, s_{i-1})}$, that is, $p_{(s_1, \dots, s_i)} \geq 0$ for all i . \square

The following result is our main tool.

Theorem 3.4 *Let p be a pebble distribution on G_ω and $x \in V(G)$. The following are equivalent.*

1. Vertex x is reachable from p .
2. There is a multiset S of pebbling moves with $p_S \geq 0$ and $p_S(x) \geq 1$.
3. There is an acyclic multiset R of pebbling moves with $p_R \geq 0$ and $p_R(x) \geq 1$.
4. Vertex x is reachable from p through a regular pebbling sequence.

Proof: If x is reachable from p then there is a sequence s of pebbling moves such that s is executable from p and $p_s(x) \geq 1$. If S is the multiset of the moves of s then $p_S \geq 0$ and $p_S(x) \geq 1$ and so (1) implies (2).

By Proposition 3.1, (2) implies (3) and by Proposition 3.3, (3) implies (4). It is clear that (4) implies (1). \square

It is convenient to write the condition $p_S \geq 0$ and $p_S(x) \geq 1$ compactly as $p_S \geq 1_{\{x\}}$ using the indicator function of the singleton set $\{x\}$.

4 Cut vertices

The pebbling number of a graph with a cut vertex often can be calculated using a simpler graph. This simplification introduces new weights. The following theorem is the main reason we study weighted graphs.

Proposition 4.1 *Let H and K be connected graphs such that $V(H) \cap V(K) = \{v\}$ and v is a cut vertex of $G = H \cup K$. Let ω be a weight function on $E(G)$. Assume that $\pi_t(K_\omega, v) = at + b$ for all t . Define a graph \tilde{G} by $V(\tilde{G}) = V(H) \dot{\cup} \{u\}$ and $E(\tilde{G}) = E(H) \cup \{vu\}$. Define a weight function on $E(\tilde{G})$ by*

$$\tilde{\omega}(e) = \begin{cases} a & \text{if } e = vu \\ \omega(e) & \text{else} \end{cases}.$$

If the goal vertex x is in $V(H)$ then $\pi(G_\omega, x) = \pi(\tilde{G}_{\tilde{\omega}}, x) + b$.

To simplify notation, we used K_ω instead of the more precise $K_{\omega|E(K)}$ even though ω is defined on values outside of $E(K)$.

Proof: The graphs are visualized in Figure 1. First we show that $\pi(G_\omega, x) \geq \pi(\tilde{G}_{\tilde{\omega}}, x) + b$. Let p be a pebble distribution on \tilde{G} with $\|p\| = \pi(G_\omega, x) - b$. We

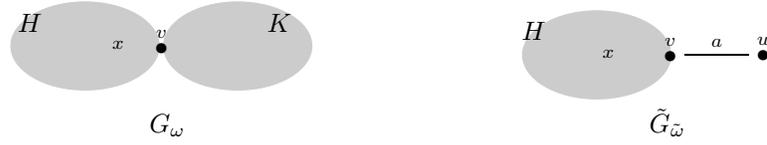


Figure 1: Simplification using the cut vertex v . If $\pi_t(K_\omega, v) = at + b$ then $\pi(G_\omega, x) = \pi(\tilde{G}_\omega, x) + b$.

create a new distribution q on G consisting of red and green pebbles. The red pebbles are placed on H exactly the same way as the pebbles in p are placed on H . The number of green pebbles is $p(u) + b$. The green pebbles are placed on K so that the number of pebbles that can be moved to v using only green pebbles is as small as possible. This minimum number is clearly $\lfloor p(u)/a \rfloor$. Note that q can place both red and green pebbles on v . Then $\|q\| = \pi(G_\omega, x)$, so there is an acyclic multiset S of pebbling moves on G_ω such that $q_S \geq 1_{\{x\}}$. Let S_H and S_K contain the moves of S inside H and K respectively so that $S = S_H \dot{\cup} S_K$.

We are going to see that moving red pebbles from H to $K \setminus \{v\}$ is not beneficial and so these moves can be eliminated. Since S is acyclic, any maximal walk in $T(K, S_K)$ starting at v is actually a path. Let us remove the pebbling moves corresponding to such maximal walks from S_K until we eliminate all walks from $T(K, S_K)$ starting at v . The choice of these maximal walks is not unique and they can overlap, so we need to eliminate them one by one in an arbitrary order. The resulting multiset $S'_K \subseteq S_K$ is balanced with q and $q_{S'_K}(v) \leq q_{S'_K}(v)$.

Executing S'_K from q cannot move more than $\lfloor p(u)/a \rfloor$ pebbles to v since S'_K does not have any effect on the red pebbles. Let R be the multiset containing the elements of S_H together with $\lfloor \|p(u)\|/a \rfloor$ copies of the move $(u \rightarrow v)$. Then it is clear that $p_R(u) \geq 0$ and p_R is not smaller than q_S on H , hence $p_R \geq 1_{\{x\}}$. Thus x is reachable from p by Theorem 3.4.

Now we show that $\pi(G_\omega, x) \leq \pi(\tilde{G}_\omega, x) + b$. Let p be a pebble distribution on G with $\|p\| = \pi(\tilde{G}_\omega, x) + b$. Let $c = \|p|_{V(K) \setminus \{v\}}\|$ be the number of pebbles in this distribution on $V(K) \setminus \{v\}$. We create a new distribution q on \tilde{G} such that q and p are the same on $V(H)$ and $q(u) = \max\{0, c - b\}$. Then $\|q\| \geq \pi(\tilde{G}_\omega, x)$ so there is an acyclic multiset S of pebbling moves on \tilde{G}_ω such that $q_S \geq 1_{\{x\}}$. There is a multiset R_1 of pebbling moves on K using only the pebbles on $V(K) \setminus \{v\}$ such that $p_{R_1}(v) = p(v) + \lfloor \max\{0, c - b\}/a \rfloor$. Let R be the multiset containing the elements of R_1 together with the elements of S different from $(u \rightarrow v)$. Then p_R is not smaller than q_S on H and p_R is nonnegative on $V(K) \setminus \{v\}$, hence $p_R \geq 1_{\{x\}}$. Thus x is reachable from p . \square

Note 4.2 *The previous proposition is applicable in many situations since the function $t \mapsto \pi_t(G, x)$ is often linear; for example for trees, complete graphs and hypercubes. In particular it is linear for cycles [11] where $\pi_t(C_{2n}) = t2^n$ and $\pi_t(C_{2n+1}) = 1 + (t - 1)2^n + 2 \lfloor \frac{2^{n+1}}{3} \rfloor$.*

Example 4.3 *The simplest nonlinear example is the wheel graph W_5 with 5 vertices and 4 spikes. It is easy to see that if x is a degree 3 vertex then*

$$\pi_t(W_5, x) = \begin{cases} 5 & \text{if } t = 1 \\ 4t & \text{if } t \geq 2 \end{cases}.$$

The behavior can be more complicated. If G is the complete graph with 7 vertices with one missing edge xy then one can verify that

$$\pi_t(G, x) = \begin{cases} 2t + 5 & \text{if } t \in \{1, 2\} \\ 4t & \text{if } t \geq 3 \end{cases}.$$

5 Simplifications using leaves

Proposition 5.1 *Let G_ω be a weighted star graph with center x and spikes xv_1, \dots, xv_n . Suppose that $a = \omega(xv_1)$ is the maximum value of ω . Then $\pi_t(G_\omega, x) = ta + \sum_{i=2}^n (\omega(xv_i) - 1)$.*

Proof: The maximum number of pebbles we can place on v_i so that at most t_i pebbles can be moved from v_i to x is $(t_i + 1)\omega(xv_i) - 1$. So

$$\begin{aligned} \pi_t(G_\omega, x) &= \max \left\{ \sum_{i=1}^n ((t_i + 1)\omega(xv_i) - 1) \mid t_1 + \dots + t_n < t \right\} + 1 \\ &= (t - 1 + 1)\omega(xv_1) - 1 + \sum_{i=2}^n ((0 + 1)\omega(xv_i) - 1) + 1 \\ &= ta + \sum_{i=2}^n (\omega(xv_i) - 1) \end{aligned}$$

since the maximum is taken when $t_1 = t - 1$ and $t_2 = \dots = t_n = 0$. □

The reader can easily verify the following result.

Proposition 5.2 *Let x, v_1 and v_2 be the consecutive vertices of the graph $G = P_3$ with weight function ω . Then $\pi_t(G_\omega, x) = t\omega(xv_1)\omega(v_1v_2)$.*

The pebbling number of a tree can be found quickly using a maximum path decomposition [4]. This method could be extended to weighted trees as well. Another approach that illustrates our simplification process is to use Propositions 4.1, 5.1 and 5.2 to simplify a weighted tree to a single edge. The process is shown in the next example.

Example 5.3 *Figure 2 shows the stages of the simplification of a tree. First we let K be the subgraph of G generated by $\{v_3, v_4, v_5\}$. Then $\pi_t(K, v_3) = 4t$ by Proposition 5.2 so we replace K by the weighted edge v_3u_1 to get $G_{\omega_1}^{(1)}$. Next we let K be the subgraph of $G_{\omega_1}^{(1)}$ generated by $\{v_1, v_2, u_1\}$. Then $\pi_t(K, v_3) = 4t + 2$ by Proposition 5.1 so we replace K by the weighted edge v_3u_2 to get $G_{\omega_2}^{(2)}$. Finally we use Proposition 5.2 again to get $G_{\omega_3}^{(3)}$.*

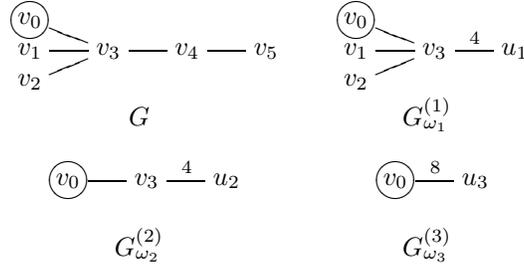


Figure 2: Simplification of a tree: $\pi(G, v_0) = \pi(G_{\omega_1}^{(1)}, v_0) = \pi(G_{\omega_2}^{(2)}, v_0) + 2 = \pi(G_{\omega_3}^{(3)}, v_0) + 2 = 10$. Unlabeled edges have weight 2.

6 Simplification using ears

In this section we use the existence of special paths in our graph to simplify the calculation of the pebbling number. A *thread* of a graph is a path whose vertices all have degree 2.

Definition 6.1 *Let x be a goal vertex in G . Let v_1, \dots, v_n be the consecutive vertices of a maximal thread T not containing x . There are unique vertices v_0 and v_{n+1} outside of T that are adjacent to v_1 and v_n respectively. The subgraph E induced by v_0, \dots, v_{n+1} is called an ear. The vertices of T are called the inner vertices of E . If $v_0 = v_{n+1}$ then E is called a closed ear. If the vertices of T are cut vertices then E is called a cut ear. If E is neither a closed ear nor a cut ear then it is called an open ear.*

Note that an ear has at least two edges. Also note that the goal vertex can be an end vertex of an ear.

6.1 Closed ears

If a closed ear has default weights then it can be replaced by a weighted edge using Proposition 4.1 and Note 4.2. The simplification is shown in Figure 3. If the closed ear has $2n$ vertices then $\pi(G_\omega, x) = \pi(\tilde{G}_\omega, x)$. If the closed ear has $2n + 1$ vertices then $\pi(G_\omega, x) = \pi(\tilde{G}_\omega, x) + 1 - 2^n + 2 \left\lfloor \frac{2^{n+1}}{3} \right\rfloor$. The edge weight is $a = 2^n$ in both cases.

6.2 Cut ears

Cut ears can be replaced by weighted edges as well. First we need the following result.

Lemma 6.2 *Let p be a maximum size pebble distribution from which the goal vertex x is not reachable. Then p has no pebbles on the inner vertices of a cut ear.*

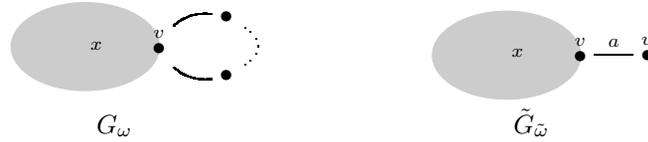


Figure 3: Substitution for a closed ear. The edge weight a is $2^{\lfloor \frac{k}{2} \rfloor}$ where k is the number of vertices of the closed ear.



Figure 4: Substitution for a cut ear. The edge weight is $\tilde{\omega}(vu) = a = 2^{n-1}$ where n is the number vertices of the path connecting v to u in G .

Proof: Suppose u is an inner vertex of the cut ear E and $p(u) > 0$. Let H and K be the connected components of $G \setminus \{u\}$ such that $x \in H$. There is a unique vertex $v \in K$ that is adjacent to u . The size of $q = p_{(v \rightarrow u)^{-1}}$ is larger than the size of p . We show that x is not reachable from q which is a contradiction.

Suppose x is reachable from q , that is, there is an acyclic multiset S of pebbling moves with $q_S \geq 1_{\{x\}}$. If $(v \rightarrow u) \in S$ then with $R = S \setminus \{(v \rightarrow u)\}$ we have $p_R = q_S \geq 1_{\{x\}}$ which is not possible. So we can assume that $(v \rightarrow u) \notin S$. Let R contain those moves of S that do not involve any edge in K . Then $p_R(u) \geq q_S(u) + 1$, $p_R(w) = q_S(w)$ for $w \in V(H)$ and $p_R(w) = p(w)$ for $w \in V(K)$. So $p_R \geq 1_{\{x\}}$ which is again impossible. \square

Proposition 6.3 *Let E be a cut ear of G_ω with end vertices v and u . Let \tilde{G} be the graph created from G by removing the inner vertices of E and adding the edge vu . Define $\tilde{\omega}$ on $E(\tilde{G})$ by*

$$\tilde{\omega}(e) = \begin{cases} a & \text{if } e = vu \\ \omega(e) & \text{else} \end{cases}$$

where a is the product of the weights of the edges of E . If the goal vertex x is not an inner vertex of E then $\pi(G_\omega, x) = \pi(\tilde{G}_{\tilde{\omega}}, x)$.

Proof: Without loss of generality we can assume that x is closer to v than to u as shown in Figure 4. Let $u = v_0, v_1, \dots, v_{n+1} = v$ be the consecutive vertices of E .

First we show that $\pi(G_\omega, x) \leq \pi(\tilde{G}_{\tilde{\omega}}, x)$. For a contradiction, assume that $\pi(G_\omega, x) > \pi(\tilde{G}_{\tilde{\omega}}, x)$. Let p be a maximum size pebble distribution on G from

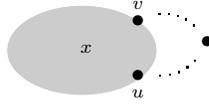


Figure 5: A graph with an open ear.

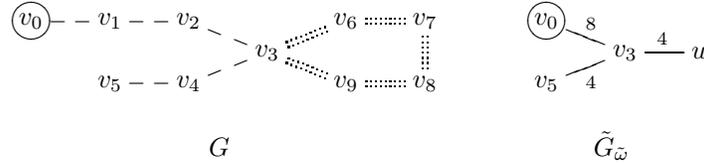


Figure 6: Simplification of G with goal vertex v_0 such that $\pi(G, v_0) = \pi(\tilde{G}_{\tilde{\omega}}, v_0) + 1 = 36$. The two cut ears denoted by dashed edges are replaced by the weighted edges v_0v_3 and v_5v_3 in \tilde{G} . The closed ear denoted by double dotted edges is replaced by the weighted edge v_3u .

which x is not reachable. By Lemma 6.2, p has no pebbles on the inner vertices of E so the restriction $q = p|_{V(\tilde{G})}$ is a pebble distribution on \tilde{G} with $\|q\| = \|p\| = \pi(G_{\omega}, x) - 1 \geq \pi(\tilde{G}_{\tilde{\omega}}, x)$. Hence there is a multiset S of pebbling moves on $\tilde{G}_{\tilde{\omega}}$ such that $q_S \geq 1_{\{x\}}$. Let R be the multiset of pebbling moves containing the moves in S with each move of the form $(u \rightarrow v)$ replaced by the moves $(v_0 \rightarrow v_1), \dots, (v_n \rightarrow v_{n+1})$. Then $q_R \geq 1_{\{x\}}$ which is a contradiction.

Now we show that $\pi(G_{\omega}, x) \geq \pi(\tilde{G}_{\tilde{\omega}}, x)$. Let p be a pebble distribution on \tilde{G} with size $\pi(G_{\omega}, x)$. Let q be the extension of p to $V(G)$ such that q is zero on the inner vertices of E . Then $\|q\| = \|p\| = \pi(G_{\omega}, x)$ and so there is an acyclic multiset S of pebbling moves such that $q_S \geq 1_{\{x\}}$. We create a multiset R of pebbling moves on \tilde{G} as follows. We start with S . We then search for a directed path in $T(G, S)$ connecting u to v and we remove all the moves corresponding to the arrows of this directed path. We do this until there are no more such directed paths. Then we add as many copies of $(u \rightarrow v)$ as the number of directed paths removed. Finally we remove all moves involving inner vertices of E . It is easy to see that $p_R \geq 1_{\{x\}}$. \square

6.3 Open ears

Figure 5 depicts an open ear. An open ear cannot be replaced by a single edge but we can still take advantage of it using squishing as explained in Section 9.

6.4 Examples

Figures 6 and 7 show two examples of simplified graphs using ears. The graph G is the same in both examples but the ears are different because the goal vertices

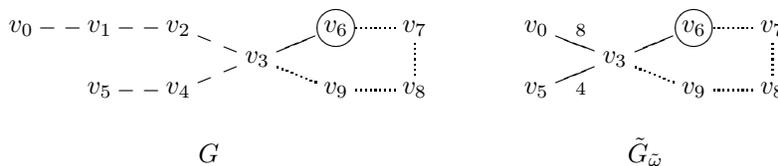


Figure 7: Simplification of G with goal vertex v_6 such that $\pi(G, v_6) = \pi(\tilde{G}_\omega, v_6) = 22$. The two cut ears denoted by dashed edges are replaced by the weighted edges v_0v_3 and v_5v_3 in \tilde{G} . The open ear denoted by dotted edges remains in \tilde{G} .

are different. In both of these examples a further simplification is possible using leaves and Proposition 5.1. The pebbling number of the graph is $\pi(G) = 36$.

The path connecting v_3 to v_5 can be simplified in two steps using leaves but we simplified it in one step using a cut ear.

Note that in the second example the end vertices v_3 and v_6 of the open ear are adjacent. This possibility is important to keep in mind during the development of an algorithm to find open ears.

7 Barely sufficient pebble distributions

Let $\mathcal{D}(G)$ be the set of pebble distributions on the graph G . For $p, q \in \mathcal{D}(G)$ we write $p \leq q$ if $p(v) \leq q(v)$ for all $v \in G$. This gives a partial order on $\mathcal{D}(G)$. We write $p < q$ if $p \leq q$ but $p \neq q$. It is clear that if a goal vertex is reachable from p and $p \leq q$ then the goal vertex is also reachable from q .

Definition 7.1 *Let x be a goal vertex of G_ω . A pebble distribution p is sufficient for x if x is reachable from p . The set of sufficient distributions for x is denoted by $\mathcal{S}(G_\omega, x)$. A pebble distribution $p \in \mathcal{S}(G_\omega, x)$ is barely sufficient for x if x is not reachable from any pebble distribution q satisfying $q < p$. The set of barely sufficient distributions for x is denoted by $\mathcal{B}(G_\omega, x)$. The set of insufficient distributions for x is $\mathcal{I}(G_\omega, x) = \mathcal{D}(G) \setminus \mathcal{S}(G_\omega, x)$. We are going to use the notation $\mathcal{S}(x)$, $\mathcal{B}(x)$ and $\mathcal{I}(x)$ if G and ω are clear from the context.*

We can partition $\mathcal{B}(x)$ into the disjoint union $\mathcal{B}_0(x) \dot{\cup} \dots \dot{\cup} \mathcal{B}_k(x)$ where $\mathcal{B}_i(x)$ contains those distributions in $\mathcal{B}(x)$ from which x is reachable in i pebbling moves but x is not reachable in fewer than i moves. Note that the only element of $\mathcal{B}_0(x)$ is the pebble distribution $1_{\{x\}}$ that contains a single pebble on x .

Example 7.2 *Figure 8 shows an example of $\mathcal{B}(G, x)$.*

The following result is the main reason for our interest in barely sufficient distributions.

Proposition 7.3 *We have $p \in \mathcal{I}(G_\omega, x)$ if and only if $q \not\leq p$ for each $q \in \mathcal{B}(G_\omega, x)$.*

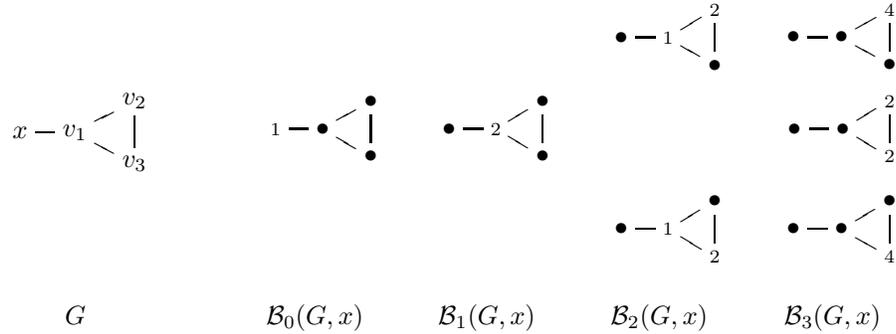


Figure 8: The barely sufficient pebble distributions for vertex x . The vertices denoted by bullets have no pebbles.

Proof: If x is reachable from a pebble distribution p then we can remove pebbles from p one by one if needed until we get a $q \in \mathcal{B}(G_\omega, x)$ that satisfies $q \leq p$. The other direction of the result is obviously true. \square

The following example shows how Proposition 7.3 can be used to find the insufficient distributions.

Example 7.4 In Example 7.2 the maximal elements of $\mathcal{I}(G, x)$ are $p(x, v_1, v_2, v_3) = (0, 1, 1, 1)$, $q(x, v_1, v_2, v_3) = (0, 0, 3, 1)$ and $r(x, v_1, v_2, v_3) = (0, 0, 1, 3)$. The maximum size is $\|q\| = 4 = \|r\|$ and so $\pi(G, x) = 5$.

Our purpose now is to construct algorithms for finding $\mathcal{B}(G_\omega, x)$ and $\pi(G_\omega, x)$.

8 Finding barely sufficient distributions

The following result shows how a superset of $\mathcal{B}(G_\omega, x)$ can be constructed using recursion starting at $\mathcal{B}_0(G_\omega, x) = \{1_{\{x\}}\}$.

Proposition 8.1 If $p \in \mathcal{B}_{i+1}(G_\omega, x)$ then $p = q_{r-1}$ for some $q \in \mathcal{B}_i(G_\omega, x)$ and pebbling move r .

Proof: Suppose that $p \in \mathcal{B}_{i+1}(G_\omega, x)$. Then there is an executable sequence $s = (s_1, \dots, s_{i+1})$ of pebbling moves such that $p_{s_i}(x) \geq 1$. Then with $q = p_{s_1}$ we clearly have $p = q_{s_1^{-1}}$. Vertex x is reachable from q in the i moves of the sequence (s_2, \dots, s_{i+1}) . If x is reachable from q in j moves then it is reachable from p in $j + 1$ moves. So x cannot be reached from q in fewer than i moves, which means that $q \in \mathcal{B}_i(G_\omega, x)$. \square

We do not have to use every pebbling move r during the construction of $\mathcal{B}_{i+1}(G_\omega, x)$ from $\mathcal{B}_i(G_\omega, x)$, as shown in the next result that essentially is a simple case of the No-Cycle Lemma.

Proposition 8.2 *Let $p \in \mathcal{S}(G_\omega, x)$. If $p_S \geq 1_{\{x\}}$ and $(v \rightarrow u) \in S$ then $q = p_{(u \rightarrow v)^{-1}} \notin \mathcal{B}(G_\omega, x)$.*

Proof: Let $\tilde{q}(u, v, *) = (q(u) - 1, q(v) - 1, q(*))$ and $R = S \setminus \{(v \rightarrow u)\}$. Then $\tilde{q} < q$ and

$$\begin{aligned} \tilde{q}_R(u, v, *) &= (\tilde{q}_S(u) - 1, \tilde{q}_S(v) + 2, \tilde{q}_S(*)) \\ &= (q_S(u) - 2, q_S(v) + 1, q_S(*)) \\ &= (p_S(u), p_S(v), p_S(*)) = p_S(u, v, *) \end{aligned}$$

which means $\tilde{q}_R = p_S \geq 1_{\{x\}}$. So q is not barely sufficient. □

An important interpretation of this result is that every distribution in $\mathcal{B}(G_\omega, x)$ can be obtained as p_T where $p = 1_{\{x\}}$ and T is a multiset of inverse pebbling moves such that $(v \rightarrow u)^{-1}$ and $(u \rightarrow v)^{-1}$ are not in T together for any u and v . Keeping track of the directions of the inverse pebbling moves speeds up the calculation of finding $\mathcal{B}(G_\omega, x)$. It also helps to eliminate moves that cannot be initial moves. If we avoid moves that are not initial moves then we automatically avoid cycles in the transition digraph.

Given a graph G , let \vec{G} be the directed graph whose vertex set is $V(G)$ and whose arrow set contains two arrows (u, v) and (v, u) for every edge $uv \in E(G)$.

Algorithm 8.3 *The algorithm shown in Figure 9 finds the set of barely sufficient distributions.*

The heart of the algorithm is Proposition 8.1. We apply inverse pebbling moves to transfer pebbles in barely sufficient distributions in hope of finding new barely sufficient distributions. We use triples of the form (p, E, W) that satisfy the following conditions:

1. x is reachable from p ;
2. if $(v, u) \notin E$ then $p_{(v \rightarrow u)^{-1}}$ is not barely sufficient;
3. if $v \in W$ then $(v \rightarrow u)$ is not an initial move of $p_{(v \rightarrow u)^{-1}}$.

The role of E is to keep track of the direction of the pebble flow so that we can avoid the back and forth transfer as explained in Proposition 8.2. The role of W is to avoid pebbling sequences that are not regular as explained in Theorem 3.4(4). Now we give the detailed explanation of the algorithm:

- lines 1–2: We fill the queue Q of barely sufficient distribution candidates with $\mathcal{B}_0(G_\omega, x)$. We set $E = E(\vec{G})$ since the pebbles can flow in any direction. We set $W = \emptyset$ since no vertex is ruled out as the starting vertex of an initial move.
- line 3: This loop takes an element (p, E, W) of Q and applies a possible inverse pebbling move to create a new distribution q . If $p \in \mathcal{B}_i(x)$ then q

Input: G_ω, x
Output: $\mathcal{B}(G_\omega, x)$

```

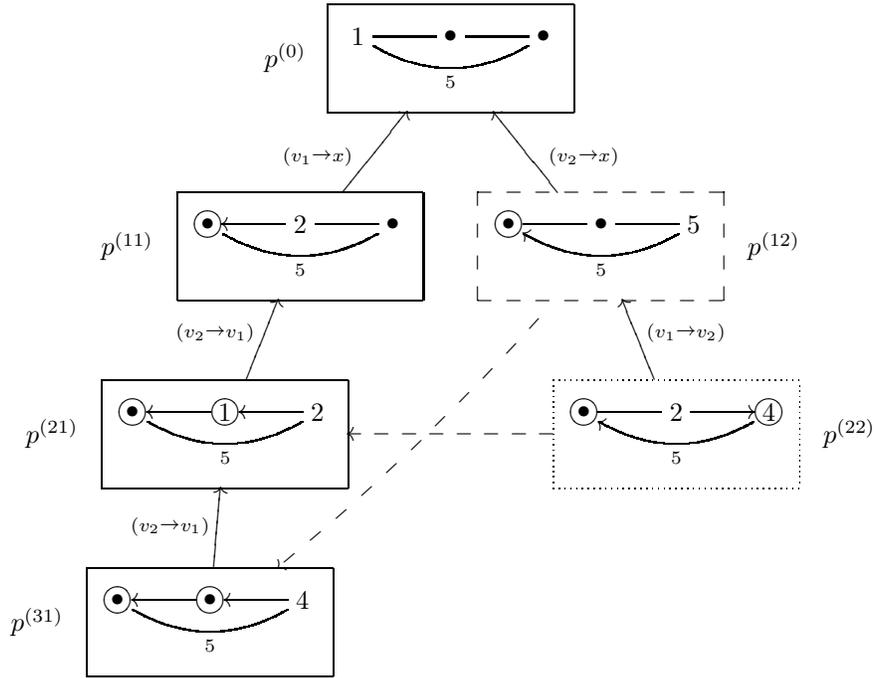
1  $(p, E, W) := (1_{\{x\}}, E(\vec{G}), \emptyset)$  // (distribution, transfers, forbidden
   vertices)
2  $Q.\text{pushBack}((p, E, W))$  // growing queue of distributions
3 foreach  $(p, E, W) \in Q$  do
4   for  $u \in \text{supp}(p)$  do //  $u$  has a pebble
5     for  $(v, u) \in E$  and  $v \notin W$  do // allowed transfer from  $v$  to  $u$ 
6        $q := p_{(v \rightarrow u)^{-1}}$  // candidate distribution
7        $F := E \setminus \{(u, v)\}$  // backward transfer no longer allowed
8        $X := W \cup \{u\}$  // transfer from  $u$  no longer allowed
9       for  $(\tilde{q}, \tilde{F}, \tilde{X}) \in Q$  do
10        if  $\tilde{q} < q$  then // candidate too large?
11          break // candidate fails
12        if  $\tilde{q} = q$  then // candidate already in queue?
13           $\tilde{F} := \tilde{F} \cap F$  // fewer allowed edges for  $\tilde{q}$ 
14           $\tilde{X} := \tilde{X} \cap X$  // not initial in any way
15          break
16        if  $\tilde{q} > q$  then //  $\tilde{q}$  is not barely sufficient?
17           $Q.\text{remove}((\tilde{q}, \tilde{F}, \tilde{X}))$  // remove  $\tilde{q}$  from queue
18        if did not break then
19           $Q.\text{pushBack}((q, F, X))$  // candidate works
20  $\mathcal{B}(G_\omega, x) := \{p \mid (p, E, W) \in Q\}$  // modification (see Note 9.3)

```

Figure 9: Algorithm to find the set $\mathcal{B}(G_\omega, x)$ of barely sufficient distributions.

is a candidate for $\mathcal{B}_{i+1}(x)$. A successful candidate q is used to create an appropriate new triple (q, F, X) . The new triple is then added at the end of Q . The loop goes through the elements of Q in order, starting from the beginning of the queue until it reaches the end of the growing queue. The loop ends eventually since we know that there are only finitely many barely sufficient distributions.

- line 4: We find a vertex u that has at least one pebble. We plan to remove a pebble from this vertex and add two pebbles to an adjacent vertex v .
- line 5: We only want to apply $(v \rightarrow u)^{-1}$ if $(u \rightarrow v)^{-1}$ was not used before and if $(v \rightarrow u)$ is an initial move. This restriction helps avoiding cycles in the transition digraph.
- line 6: We apply the inverse pebbling move $(v \rightarrow u)^{-1}$ to create the new barely sufficient candidate q .
- line 7: According to Proposition 8.2, we do not want to apply $(u \rightarrow v)^{-1}$ since we already used $(v \rightarrow u)^{-1}$.
- line 8: Any move of the form $(u \rightarrow w)$ is not an initial move since we already have a move of the form $(v \rightarrow u)$.
- line 9: The loop checks the newly created candidate against the other distributions in the queue.
- lines 10–11: We already put a smaller candidate in the queue so the new candidate cannot be barely sufficient.
- line 12: The new candidate q is already in the queue. It is likely that it was created using different inverse pebbling moves. We do not add this candidate to the queue twice. Still, we can update the information about this distribution in the queue by updating \tilde{F} and \tilde{X} .
- line 13: We can reduce the possible inverse pebbling moves using Proposition 8.2. A distribution is not barely sufficient if the goal vertex is reachable from this distribution using any pebbling sequence that is not acyclic. So we do not need to consider any inverse move that produces such a distribution.
- line 14: It is possible that a move is initial in one set of pebbling moves but not in another set. We only want to declare a move not initial if it is not initial in every possible set of pebbling moves that reaches the goal vertex. So we only keep a vertex v in \tilde{X} if $(v \rightarrow u)$ is not a possible initial move in the new triple (q, F, X) either.
- lines 16–17: If the new candidate is smaller than a distribution \tilde{q} in the queue then \tilde{q} cannot be barely sufficient. Therefore we remove it from the queue.



- | | |
|--|---|
| <ol style="list-style-type: none"> 1. $\{p^{(0)}\}$ 2. $\{p^{(0)}, p^{(11)}\}$ 3. $\{p^{(0)}, p^{(11)}, p^{(12)}\}$ 4. $\{p^{(0)}, p^{(11)}, p^{(12)}, p^{(21)}\}$ | <ol style="list-style-type: none"> 5. $p^{(22)}$ tested but is larger than $p^{(21)}$ 6. $p^{(31)}$ tested, it is smaller than $p^{(12)}$ 7. $\{p^{(0)}, p^{(11)}, p^{(21)}\}$ 8. $\{p^{(0)}, p^{(11)}, p^{(21)}, p^{(31)}\}$ |
|--|---|

Figure 10: Demonstration of Algorithm 8.3. The distributions in solid frames belong to $\mathcal{B}(G_\omega, x)$. Distribution $p^{(22)}$ in a dotted frame is never in the queue. A solid arrow from q to p is drawn with label $(v \rightarrow u)$ if $q = p_{(v \rightarrow u)^{-1}}$. A dashed arrow from q to p is drawn if $q \geq p$ and so $q \notin \mathcal{B}(G_\omega, x)$. The circled vertices belong to W . The table shows how the set of distributions in queue Q changes during the execution of the algorithm.

- lines 18–19: The new candidate is added to the queue.

Example 8.4 Let x be the goal vertex and v_1 and v_2 be the other vertices of the complete graph $G = K_3$ and let $\omega(xv_2) = 5$. Figure 10 shows how Algorithm 8.3 finds $\mathcal{B}(G_\omega, x) = \{p^{(0)}, p^{(11)}, p^{(21)}, p^{(31)}\}$. Note that $p^{(12)}$ is added to the queue and only removed later when $p^{(31)}$ is found. This late recognition of the fact that $p^{(12)}$ is not barely sufficient is the reason why the algorithm needs to test $p^{(22)}$ as a candidate.

9 Squished distributions

In this section we prove a version of the Squishing Lemma of [2] using open ears. A pebble distribution is *squished* on a thread P if all the pebbles on P are placed on a single vertex of P or on two adjacent vertices of P . A pebble distribution can be made squished on a thread as shown in the proof of the next result.

Lemma 9.1 (*Squishing*) *If vertex x is not reachable from a pebble distribution p with size n , then there is a pebble distribution of size n that is squished on each unweighted open ear and from which x is still not reachable.*

Proof: Let E be an unweighted open ear with consecutive vertices v_0, \dots, v_n . Suppose that the pebble distribution p is not squished on E . Let i be the smallest and j be the largest index for which $p(v_i) > 0$ and $p(v_j) > 0$. Note that we must have $j - i \geq 2$. Define a new pebble distribution q by applying a *squishing move* that moves one pebble from v_i to v_k and another pebble from v_j to v_k for some k satisfying $i < k < j$.

Suppose x is reachable from q , that is, there is an acyclic multiset S of pebbling moves such that $q_S \geq 1_{\{x\}}$. Pick a maximal directed path of $T(G, S)$ with consecutive vertices $v_k = w_0, w_1, \dots, w_l$ all in the set $\{v_i, v_{i+1}, \dots, v_j\}$. Let D be the set of moves corresponding to the arrows of this directed path, that is, $D = \{(w_0 \rightarrow w_1), \dots, (w_{l-1} \rightarrow w_l)\}$ and let $R = S \setminus D$. We need to consider three cases depending on whether $w_l = v_k$, $w_l \in \{v_i, v_j\}$ or $w_l \notin \{v_k, v_i, v_j\}$. It is easy to see that in all three cases we must have $p_R \geq 1_{\{x\}}$ which is a contradiction.

Applying squishing moves repeatedly on E makes the pebble distribution squished on E . This procedure keeps the goal vertex x unreachable. A squishing move on E might remove a pebble from another open ear but it cannot add a pebble to it. So if the distribution is squished on an open ear then it remains squished after the application of a squishing move on E . So the desired pebble distribution can be reached by applying all the available squishing moves on all unweighted ears in any order. \square

The set $\mathcal{I}^s(G_\omega, x)$ of *squished insufficient distributions* is the set of those elements of $\mathcal{I}(G_\omega, x)$ that are squished on all open ears of G . The set $\mathcal{B}^s(G_\omega, x)$ of *squished barely sufficient distributions* is the set of those elements of $\mathcal{B}(G_\omega, x)$ that are squished on all open ears of G .

Proposition 9.2 *Let p be squished. We have $p \in \mathcal{I}^s(G_\omega, x)$ if and only if $q \not\leq p$ for each $q \in \mathcal{B}^s(G_\omega, x)$.*

Proof: The result follows from Proposition 7.3. \square

Note 9.3 *We can find $\mathcal{B}^s(G_\omega, x)$ by a slight modification of Algorithm 8.3. On line 20 we remove (p, E, W) from Q if p is not squished. This must be done at the end, since we may not be able to find all the squished distributions if we only considered squished distributions during the algorithm.*

Corollary 9.4 $\pi(G_\omega, x) = \max\{\|p\| : p \in \mathcal{I}^s(G_\omega, x)\} + 1$.

Proof: The result follows from the Squishing Lemma using $\pi(G_\omega, x) = \max\{\|p\| : p \in \mathcal{I}(G_\omega, x)\} + 1$. \square

10 Finding insufficient distributions

Now we present an algorithm for finding $\mathcal{I}^s(x)$. Let \bar{p} be the pebble distribution defined by $\bar{p}(v) = \max\{q(v) \mid q \in \mathcal{B}^s(x)\}$ for all $v \in V(G)$. It is easy to see that $\bar{p}(v) = 2^{\text{dist}(v,x)}$ for all v where $\text{dist}(v,x)$ is the weighted distance between v and x . It is clear that if $p \in \mathcal{I}^s(x)$ then $p \leq \bar{p}$. The idea of the algorithm is to decrease the number of pebbles at certain vertices of \bar{p} until it becomes insufficient.

Algorithm 10.1 *The algorithm shown in Figure 11 finds $\pi(G_\omega, x)$ using $\mathcal{B}^s(G_\omega, x)$.*

The algorithm uses Proposition 9.2 and Corollary 9.4. We wish to maximize the size of a squished distribution p subject to the constraint that for each q in $\mathcal{B}^s(G_\omega, x)$, some vertex v satisfies $p(v) < q(v)$. The input $\mathcal{B}^s(G_\omega, x)$ is the output of the modified Algorithm 8.3 as explained in Note 9.3. It contains all the squished barely sufficient distributions. Now we give the detailed explanation of the algorithm:

- lines 1–2: We find the upper bound \bar{p} for $\mathcal{I}(G_\omega, x)$. Every insufficient distribution can be constructed from \bar{p} by decreasing the number of pebbles on some vertices.
- line 3: Distribution \bar{p} might not be squished. So instead of using \bar{p} we can use a new distribution p by removing all the pebbles from \bar{p} at a few vertices until the distribution becomes squished. Every squished insufficient distribution can be constructed from one such p by decreasing the number of pebbles on some vertices. We use the temporary ordered queue P to keep candidates for such squished p 's.
- line 4: We are not done until there are candidates in P .
- line 5: Take a candidate p from P .

Input: $\mathcal{C} := \mathcal{B}^s(G_\omega, x)$
Output: $\pi(G_\omega, x)$

```

1 for  $v \in V(G)$  do
2    $\bar{p}(v) := \max\{q(v) \mid q \in \mathcal{C}\}$  //  $\bar{p}$  is an upper bound for  $\mathcal{I}^s(G_\omega, x)$ 
3  $P.\text{pushBack}(\bar{p})$  // ordered queue of not yet squished candidates
4 while  $P$  not empty do // more candidates to try
5    $P.\text{popBack}(p)$  // work with candidate  $p$ 
6   if  $p$  squished then
7      $Q.\text{insert}((p, 1))$  // ordered queue of squished candidates
8     continue // nothing more to do with  $p$ 
9   for  $v \in V(G)$  do // try to improve  $p$ 
10     $q := p$  // modify  $p$ 
11     $q(v) := 0$  // this might make it squished
12     $P.\text{insert}(q)$  // add improved candidate to the queue
13  $M := 0$  // size of best insufficient distribution so far
14 while  $Q$  not empty do // more candidates to try
15    $Q.\text{popBack}((p, i))$  //  $\mathcal{C}[1] \not\leq p, \dots, \mathcal{C}[i-1] \not\leq p$ 
16   if  $\|p\| \leq M$  then // too few pebbles?
17     continue // candidate has no hope to be better
18   while  $i \leq |\mathcal{C}|$  and  $\mathcal{C}[i] \not\leq p$  do // find first  $i$  such that  $\mathcal{C}[i] \leq p$ 
19      $i := i + 1$  // not found yet
20   if  $i > |\mathcal{C}|$  then // no such  $i$ , candidate is insufficient
21      $M := \|p\|$  //  $p$  is the best insufficient distribution so far
22     continue // nothing more to do with  $p$ 
23   for  $v \in V(G)$  do // find ways to satisfy  $\mathcal{C}[i] \leq q$ 
24      $q := p$  // modify  $p$ 
25      $q(v) := \mathcal{C}[i](v) - 1$  // enforce  $\mathcal{C}[i] \leq q$ 
26     if  $q(v) \geq 0$  then // nonnegative number of pebbles on  $v$ ?
27        $Q.\text{insert}((q, i + 1))$  // add improved candidate to the
28       queue
29  $\pi(G_\omega, x) := M + 1$ 

```

Figure 11: Algorithm to find the distribution with the most pebbles that is insufficient for the goal vertex.

- lines 6–7: If p is squished then we insert $(p, 1)$ in the ordered queue Q . This new queue contains candidates for insufficient distributions. The insert operation uses binary search to locate the right position. It only adds the new element to the queue at this position if it is not there already.
- line 9: We try to improve p by removing all pebbles at vertex v .
- line 10: Distribution q is going to be the hopefully improved version of p .
- line 11: We remove all the pebbles at vertex v .
- line 12: We insert the new candidate q into P .
- line 13: The variable M is initialized; it will equal the maximum size $\max\{\|p\| : p \in \mathcal{I}^s(G_\omega, x)\}$ of an insufficient distribution.
- lines 14: We are finding M using Proposition 9.2 and Corollary 9.4. Queue Q contains pairs of the form (p, i) . Such a p always satisfies $q \not\leq p$ if q is any one of the first $i - 1$ elements of $\mathcal{B}^s(G_\omega, x)$. In each iteration we replace $(p, i) \in Q$ by possibly several new elements in the queue of the form $(q, i + 1)$.
- line 15: Take a candidate p from Q . We know that $\mathcal{C}[1] \not\leq p, \dots, \mathcal{C}[i - 1] \not\leq p$.
- lines 16–17: Distribution p can be discarded if it has no more than M pebbles. We already have an insufficient distribution containing M pebbles so p only has a chance for improvement if it has more than M pebbles.
- line 18–19: The loop finds the first index i for which $\mathcal{C}[i] \leq p$ or quits if no such i exists.
- line 20–21: If $i > |\mathcal{C}|$ then p is our best performing insufficient distribution so far.
- line 23: We need to remove some pebbles from p to enforce that $\mathcal{C}[i] \not\leq p$. We try to do this at each vertex.
- line 24–25: We build a new distribution q by decreasing the number of pebbles at vertex v just enough to have $\mathcal{C}[i] \not\leq q$.
- lines 26–27: If the new q is actually a distribution then we insert $(q, i + 1)$ into Q .
- line 28: The pebbling number is calculated according to Corollary 9.4.

It is important to keep our queues sorted and to use binary search at the insert operations. Without this the algorithm becomes too slow to be practical.

<ol style="list-style-type: none"> 1. $Q = \{(\langle 1, 2, 4 \rangle, 1)\}$ 2. $Q = \{(\langle 0, 2, 4 \rangle, 2)\}$ 3. $Q = \{(\langle 0, 1, 4 \rangle, 3)\}$ 4. $Q = \{(\langle 0, 0, 4 \rangle, 4), (\langle 0, 1, 1 \rangle, 4)\}$ 		<ol style="list-style-type: none"> 5. $\langle 0, 1, 1 \rangle$ is insufficient, $M = \ \langle 0, 1, 1 \rangle\$ 6. $Q = \{(\langle 0, 0, 4 \rangle, 4)\}$ 7. $Q = \{(\langle 0, 0, 3 \rangle, 5)\}$ 8. $\langle 0, 0, 3 \rangle$ is insufficient, $M = \ \langle 0, 0, 3 \rangle\$
--	--	--

Figure 12: Demonstration of Algorithm 10.1 for Example 10.2.

Example 10.2 *Now we finish the work started in Example 8.4 for the graph $G = K_3$ by showing how Algorithm 10.1 finds $\pi(G_\omega, x)$. We are going to use the temporary notation $\langle a, b, c \rangle$ for the pebble distribution $p(x, v_1, v_2) = (a, b, c)$. All the pebble distributions in $\mathcal{B}(G_\omega, x)$ are squished and so $\mathcal{B}^s(G_\omega, x) = \{\langle 1, 0, 0 \rangle, \langle 0, 2, 0 \rangle, \langle 0, 1, 2 \rangle, \langle 0, 0, 4 \rangle\}$. The distribution $\bar{p} = \langle 1, 2, 4 \rangle$ is also squished so $Q = \{(\langle 1, 2, 4 \rangle, 1)\}$ has only one element at the beginning. Figure 12 shows how queue Q changes during the execution of the algorithm. We conclude that $\pi(G_\omega, x) = M + 1 = 4$. It is easy to see that $\pi(G_\omega, v_1) = 3$ and $\pi(G_\omega, v_2) = 4$ and so $\pi(G_\omega) = 4$.*

11 Test results

We tested our algorithms by calculating the pebbling number of every connected graph with fewer than 10 vertices. We used Nauty [12] to generate these graphs and their automorphism groups. We simplified each graph as follows. For each goal vertex we replaced each closed ear and cut ear with a weighted edge as described in Subsections 6.1 and 6.2. Then we recursively used available leaves to simplify the graph as much as possible as described in Section 5. Then we ran Algorithms 8.3 and 10.1 to find the pebbling number of the simplified graph.

The automorphism group helped us reducing the number of goal vertices to representatives of orbits. It is well known that the hardest to reach goal vertex in a tree is a leaf, so in trees we only picked leaf vertices for the goal vertex.

The algorithms were coded in C++ using the Standard Template Library. The code was compiled with the gnu compiler. It took about a day on a 3 GHz Unix machine to finish the calculations. The calculation for graphs with fewer than 9 vertices took less than 10 minutes. Table 1 shows the frequency of the pebbling numbers. The result confirms the existence of gaps in the spectrum of pebbling numbers described in [3]. We checked our results on many graphs with known pebbling numbers such as paths, complete graphs, cycles, some trees, the Lemke graph, and the Petersen graph. The pebbling numbers are available on the author’s web page.

$ V(G) = 1$	6	5							
	8	2	8	113	16	190	16	2505	
1	1	9	3	9	125	17	341	17	5293
		10	129	10	129	18	333	18	5992
$ V(G) = 2$		11	68	11	68	19	148	19	4070
	16	1	12	4	12	4	20	15	20
									21
									137
2	1	$ V(G) = 6$		16	23	32	36	22	5
				17	35	33	52	23	8
$ V(G) = 3$				18	22	34	34		
	6	45	19	2	35	3		32	318
	7	15						33	626
3	1		32	4	64	6	34	579	
4	1	8	13	33	5	65	6	35	261
		9	16					36	33
$ V(G) = 4$		10	13	64	1	128	1	39	1
		11	1						
				$ V(G) = 8$		$ V(G) = 9$			
4	3	16	4					64	50
5	2	17	4	8	4494	9	126646	65	79
				9	1658	10	43935	66	47
8	1	32	1	10	1870	11	41222	67	4
$ V(G) = 5$				11	1425	12	22756		
		$ V(G) = 7$		12	478	13	4975	128	6
				13	26	14	208	129	7
5	10	7	322	14	1	15	6	256	1

Table 1: The frequency of pebbling numbers for graphs with less than 10 vertices. The data is grouped by the number $|V(G)|$ of vertices in the graph. Each data row contains a possible pebbling number followed by the frequency of this pebbling number.

12 Further questions

1. We do not have any example where $t \mapsto \pi_t(G, x)$ is not linear for $t \geq 3$. Is it true that $t \mapsto \pi_t(G, x)$ is always linear for $t \geq t_0$ for some t_0 ? What properties of the graph can be used to find t_0 ? Note that the coefficient of t in such a linear function should be the fractional pebbling number of [8].
2. Is it possible that the goal vertex x that maximizes $\pi(G, x)$ is an interior vertex of a cut ear of G ? The answer seems to be no and it may depend on the first question. We could use this result to speed up our calculation of $\pi(G)$ since we would have to test fewer goal vertices.
3. Is it possible to take advantage of open ears in a better way? Can we simplify a graph with an open ear so that it has fewer vertices? This simplification might help tremendously if it is compatible with ear decomposition. It might be possible to reduce a graph completely if we know π_t for all graphs with fewer vertices.
4. Perhaps adding extra weighted edges could speed up Algorithm 8.3 in certain cases. For example in an open ear, we could connect the end vertices to the interior vertices with appropriate weights depending on the distance.
5. What general results are there about the pebbling number of weighted graphs? In particular, Graham conjectures that $\pi(G \square H) \leq \pi(G)\pi(H)$. Can we extend this conjecture for weighted graphs? The p -pebbling version of the conjecture stated in [5] is a step in this direction.
6. What is the pebbling number of simple weighted graphs like a weighted cycle? Note that pebbling in unweighted graphs can be reduced to pebbling in complete weighted graphs by assigning huge weights to non-adjacent vertex pairs.

References

- [1] Airat Bekmetjev and Charles A. Cusack. Pebbling algorithms in diameter two graphs. *SIAM J. Discrete Math.*, 23(2):634–646, 2009.
- [2] David P. Bunde, Erin W. Chambers, Daniel Cranston, Kevin Milans, and Douglas B. West. Pebbling and optimal pebbling in graphs. *J. Graph Theory*, 57(3):215–238, 2008.
- [3] Christopher Cabanski. Forbidden pebbling numbers. University of Dayton honors thesis, 2007.
- [4] Fan R. K. Chung. Pebbling in hypercubes. *SIAM J. Discrete Math.*, 2(4):467–472, 1989.
- [5] T. A. Clarke, R. A. Hochberg, and G. H. Hurlbert. Pebbling in diameter two graphs and products of paths. *J. Graph Theory*, 25(2):119–128, 1997.
- [6] Betsy Crull, Tammy Cundiff, Paul Feltman, Glenn H. Hurlbert, Lara Pudwell, Zsuzsanna Szaniszló, and Zsolt Tuza. The cover pebbling number of graphs. *Discrete Math.*, 296(1):15–23, 2005.
- [7] Shawn Elledge and Glenn H. Hurlbert. An application of graph pebbling to zero-sum sequences in abelian groups. *Integers*, 5(1):A17, 10 pp. (electronic), 2005.
- [8] D. Herscovici, B. Hester, and G. H. Hurlbert. Diameter bounds, fractional pebbling, and pebbling with arbitrary target distributions. arXiv:0905.3949v1.
- [9] Glenn H. Hurlbert. A survey of graph pebbling. In *Proceedings of the Thirtieth Southeastern International Conference on Combinatorics, Graph Theory, and Computing (Boca Raton, FL, 1999)*, volume 139, pages 41–64, 1999.
- [10] Glenn H. Hurlbert. Recent progress in graph pebbling. *Graph Theory Notes N. Y.*, 49:25–37, 2005.
- [11] A. Lourdusamy and S. Somasundaram. The t -pebbling number of graphs. *Southeast Asian Bull. Math.*, 30(5):907–914, 2006.
- [12] Brendan D. McKay. Practical graph isomorphism. In *Proceedings of the Tenth Manitoba Conference on Numerical Mathematics and Computing, Vol. I (Winnipeg, Man., 1980)*, volume 30, pages 45–87, 1981.
- [13] Kevin Milans and Bryan Clark. The complexity of graph pebbling. *SIAM J. Discrete Math.*, 20(3):769–798 (electronic), 2006.
- [14] David Moews. Pebbling graphs. *J. Combin. Theory Ser. B*, 55(2):244–252, 1992.