Inapproximability of the Standard Pebble Game and Hard to Pebble Graphs

Inapproximability of the Standard Pebble Game and Hard to Pebble Graphs*

Erik D. Demaine[1] and Quanquan C. Liu†[1]

[1]Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology, Cambridge, MA, USA
{edemaine,quanquan}@mit.edu

Abstract

Pebble games are single-player games on DAGs involving placing and moving pebbles on nodes of the graph according to a certain set of rules. The goal is to pebble a set of target nodes using a minimum number of pebbles. In this paper, we present a possibly simpler proof of the result in [CLNV15] and strengthen the result to show that it is PSPACE-hard to determine the minimum number of pebbles to an additive $n^{1/3-\varepsilon}$ term for all $\varepsilon > 0$, which improves upon the currently known additive *constant* hardness of approximation [CLNV15] in the standard pebble game. We also introduce a family of explicit, constant indegree graphs with n nodes where there exists a graph in the family such that using constant k pebbles requires $\Omega(n^k)$ moves to pebble in both the standard and black-white pebble games. This independently answers an open question summarized in [Nor15] of whether a family of DAGs exists that meets the upper bound of $O(n^k)$ moves using constant k pebbles with a different construction than that presented in [AdRNV17].

1 Introduction

Pebble games were originally introduced to study compiler operations and programming languages. For such applications, a DAG represents the computational dependency of each operation on a set of previous operations and pebbles represent register allocation. Minimizing the amount of resources allocated to perform a computation is accomplished by minimizing the number of pebbles placed on the graph [Set75]. The *standard pebble game* (also known as the *black pebble game*) is traditionally used to model such behavior. In the standard pebble game, one is given a DAG, $G = (V, E)$, with n nodes and constant indegree and told to perform a set of *pebbling moves* that places, removes, or slides pebbles around the nodes of G.

The premise of such games is given some input modeled by *source* nodes $S \subseteq V$ one should compute some set of outputs modeled as target nodes $T \subseteq V$. In terms of G, S is typically the set of nodes without incoming edges and T is typically the set of nodes without outgoing edges. The rules of the standard pebble game are as follows:

*A preliminary version of this paper was presented at the *Algorithms and Data Structures Symposium (WADS 2017)*

†This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. (1122374).

Standard Pebble Game

Input: Given a DAG, $G = (V, E)$. Let $\mathsf{pred}(v) = \{u \in V : (u, v) \in E\}$. Let $S \subseteq V$ be the set of sources of G and $T \subseteq V$ be the set of targets of G. Let $\mathcal{P} = \{P_0, \ldots, P_\tau\}$ be a valid pebbling strategy that obeys the following rules where P_i is a set of nodes containing pebbles at timestep i and $P_0 = \emptyset$ and $P_\tau = \{T\}$. Let $\mathsf{Peb}(G, \mathcal{P}) = \max_{i \in [\tau]}\{|P_i|\}$.

Rules:

1. At most one pebble can be placed or removed from a node at a time.
2. A pebble can be placed on any source, $s \in S$.
3. A pebble can be removed from any vertex.
4. A pebble can be placed on a non-source vertex, v, at time i if and only if its direct predecessors are pebbled, $\mathsf{pred}(v) \in P_{i-1}$.
5. A pebble can slide from vertex v to vertex w at time i if and only if $(v, w) \in E$ and $\mathsf{pred}(w) \in P_{i-1}$.

Goal: Determine $\min_{\mathcal{P}}\{\mathsf{Peb}(G, \mathcal{P})\}$ using a valid strategy $\mathcal{P}$.

In addition to the standard pebble game, other pebble games are useful for studying computation. The *red-blue pebble game* is used to study I/O complexity [JWK81], the *reversible pebble game* is used to model reversible computation [Ben89], and the *black-white pebble game* is used to model non-deterministic straight-line programs [CS74]. Although we will be proving a result about the black-white pebble game in Section 4, we will defer introducing the rules of the game to the later parts of the paper since the black-white pebble game is not central to the main results of this paper.

Much previous research has focused on proving lower and upper bounds on the *pebbling space cost* (i.e. the maximum number of pebbles used at any point in time) of pebbling a given DAG under the rules of each of these games. For all of the aforementioned pebble games (except the red-blue pebble game since it relies on a different set of parameters), any DAG can be pebbled using $O(n/\log n)$ pebbles [GT78, HPV77, PTC76]. Furthermore, there exist DAGs for each of the games that require $\Omega(n/\log n)$ pebbles [GT78, HPV77, PTC76].

It turns out that finding a strategy to optimally pebble a graph in the standard pebble game is computationally difficult even when each vertex is allowed to be pebbled only once. Specifically, finding the minimum number of black pebbles needed to pebble a DAG in the standard pebble game is PSPACE-complete [GLT79] and finding the minimum number of black pebbles needed in the one-shot case is NP-complete [Set75]. In addition, finding the minimum number of pebbles in both the black-white and reversible pebble games have been recently shown to be both PSPACE-complete [CLNV15, HP10]. But the result for the black-white pebble game is proven for unbounded indegree [HP10]. A key open question in the field is whether hardness results can be obtained for constant indegree graphs for the black-white pebble game. However, whether it is possible to find good approximate solutions to the minimization problem has barely been studied. In fact, it was not known until this paper whether it is hard to find the minimum number of pebbles within even a non-constant *additive* term [CLNV15]. The best known multiplicative approximation factor is the very loose $\Theta(n/\log n)$ which is the pebbling space upper bound [HPV77], leaving much room for

improvement.

Our results deal primarily with the standard pebble game, but we believe that the techniques could be extended to show hardness of approximation for other pebble games. We prove the following:

Theorem 1. *The minimum number of pebbles needed in the standard pebble game on DAGs with maximum indegree* 2 *is PSPACE-hard to approximate to within an additive* $n^{1/3-\varepsilon}$ *for any* $\varepsilon > 0$.

In addition to determining the pebbling space cost, we sometimes also care about pebbling time which refers to the number of operations (placements, removals, or slides) that a strategy uses. For example, such a situation arises if we care not only about the memory used in computation but also the time of computation. It is previously known that there exists a family of graphs such that, given $\Theta(\frac{n}{\log n})$ pebbles, one is required to use $\Omega(2^{\Theta(\frac{n}{\log n})})$ moves to pebble any graphs with n nodes in the family [LT79].

Less is known about the trade-offs when a small number (e.g. constant k) of pebbles is used until the very recent, independent result presented in [AdRNV17]. It can be easily shown through a combinatorial argument that the maximum number of moves necessary using $k = O(1)$ pebbles to pebble n nodes is $O(n^k)$ [Nor15]. It is an open question whether it is possible to prove a time-space trade-off such that using $k = O(1)$ pebbles requires $\Omega(n^k)$ time. In this paper, we resolve this open question for both the standard pebble and the black-white pebble games using an independent construction from that presented in [AdRNV17].

Theorem 2. *There exists a family of graphs with* n *vertices and maximum indegree* 2 *such that* $\Omega(n^k)$ *moves are necessary to pebble any graph with* n *vertices in the family using constant* k *pebbles in both the standard and black-white pebble games.*

The organization of the paper is as follows. First, in Section 2, we provide the definitions and terminology we use in the remaining parts of the paper. Then, in Section 3, we provide a proof for the inapproximability of the standard pebble game to an $n^{1/3-\varepsilon}$ additive factor.

In Section 4, we present our hard to pebble graph families using $k < \sqrt{n}$ pebbles and prove that the family takes $\Omega(n^k)$ moves to pebble in both the standard and black-white pebble games when $k = O(1)$.

Finally, in Section 5, we discuss some open problems resulting from this paper.

2 Definitions and Terminology

In this section, we define the terminology we use throughout the rest of the paper. All of the pebble games we consider in this paper are played on directed acyclic graphs (DAGs), and our results are given in terms of DAGs with maximum indegree 2. We define such a DAG as $G = (V, E)$ where $|V| = n$ and $|E| = m$.

The purpose of any pebble game is to pebble a set of targets $T \subseteq V$ using minimum number of pebbles. In all pebble games we consider, a player can always place a pebble on any source node, $S \subseteq V$. Usually, S consists of all nodes with indegree 0 and T consists of all nodes with outdegree 0.

A *sequential pebbling strategy*, $\mathcal{P} = [P_0, \ldots, P_\tau]$ is a series of configurations of pebbles on G where each P_i is a set of pebbled vertices $P_i \subseteq V$. P_i follows from P_{i-1} by the rules of the game and $P_0 = \emptyset$ and $P_\tau = T$. Then, by definition, $|P_i|$ is the number of pebbles used in configuration P_i. For a *sequential* strategy, $|P_{i-1}| - 1 \leq |P_i| \leq |P_{i-1}| + 1$ for all $i \in [\tau] = [1, \ldots, \tau]$ (i.e. at most one

pebble can be placed, removed, or slid on the graph at any time). In this paper, we only consider sequential strategies.

Given any strategy $\mathcal{P}$ for pebbling G, the *pebbling space cost*, $\mathsf{Peb}(G, \mathcal{P})$, of $\mathcal{P}$ is defined as the maximum number of pebbles used by the strategy at any time: $\mathsf{Peb}(G, \mathcal{P}) = \max_{i \in [\tau]}\{|P_i|\}$.

The *minimum pebbling space cost* of G, $\mathsf{Peb}(G)$, is defined as the smallest space cost over the set of all valid strategies, $\mathbb{P}$, for G:

Definition 1 (Minimum Pebbling Space Cost)**.**

$$\mathsf{Peb}(G) = \min_{\mathcal{P} \in \mathbb{P}}\{\mathsf{Peb}(G, \mathcal{P})\}.$$

The *pebbling time cost*, $\mathsf{Time}(G, \mathcal{P}) = |\mathcal{P}| - 1$, of a strategy $\mathcal{P}$ using s pebbles is the number of moves used by the strategy. The *minimum pebbling time cost* of any strategy that has pebbling space cost s is the minimum number of moves used by any such strategy. We know that the pebbling time cost is at least N since at least n moves are necessary to place a pebble on every node in an n node DAG.

Definition 2 (Minimum Pebbling Time Cost)**.**

$$\mathsf{Time}(G, s) = \min_{\mathcal{P}' \in \{\mathcal{P} \in \mathbb{P} : |\mathcal{P}| \leq s\}}\{\mathsf{Time}(G, \mathcal{P}')\} \geq n.$$

3 Inapproximability of the Standard Pebble Game

In this section, we provide an alternative proof of the result presented in [CLNV15] that the standard pebble game is inapproximable to any constant additive factor. Then, we show that our proof technique can be used to show our main result stated in Theorem 1.

We first introduce the PSPACE-completeness proof presented by [GLT79] because we modify this proof to prove our main result.

3.1 Standard Pebbling is PSPACE-complete [GLT79]

The reduction is performed from the PSPACE-complete problem, quantified boolean formula (QBF). In an instance of QBF, we are given a quantified boolean formula of the form: $B = Q_1 x_1 \cdots Q_u x_u F$ where Q_i is either an existential or universal quantifier, each x_i is a Boolean variable, and F is an unquantified boolean formula containing variable x_i in CNF form with 3 variables per clause. The decision problem is to determine whether B is satisfiable for some assignment of truth values to the existential variables and for all truth assignments to the universal variables.

The reduction is done by constructing a graph $G = (V, E)$ with one target node, q_1. G can be pebbled with s pebbles if and only if B is satisfiable. Rather than constructing gadgets to represent each variable in F, [GLT79] constructs a gadget to represent each literal. For each $x_i \in F$, they create a gadget that can be set in one of three possible settings: $(x_i = True, \overline{x_i} = False)$, $(x_i = False, \overline{x_i} = True)$, or $(x_i = False, \overline{x_i} = False)$. Let $F(e_1, e_2, \ldots, e_i, e_{i+1}, \ldots, e_{2v-1}, e_{2v})$ represent the formula F obtained by substituting e_i for x_i and e_{i+1} for $\overline{x_i}$ for all $i \in [v]$. Note that if $F(e_1, e_2, \ldots, False, False, \ldots, e_{2v-1}, e_{2v})$ is true, then, trivially, $F(e_1, e_2, \ldots, True, False, \ldots, e_{2v-1}, e_{2v})$ and $F(e_1, e_2, \ldots, False, True, \ldots, e_{2v-1}, e_{2v})$ are also both true.

The proof of their reduction from QBF relies on the following key gadgets.

Variable Gadget: A variable gadget is created for each variable x_i, $i \in [v]$ in F with two paths, one path representing each literal. Each variable gadget can be in one of the following three configurations shown in Fig. 1.

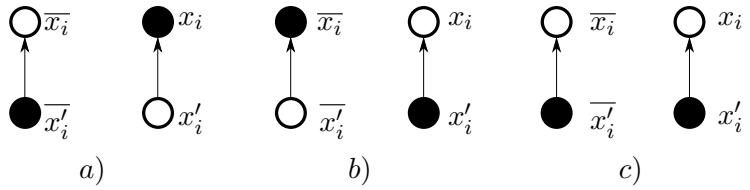

Figure 1: Variable gadget as used in [GLT79]. Each variable can be in one of the 3 configurations shown. a) True configuration. b) False configuration. c) Double false configuration. Figure recreated from [GLT79].

Universal Quantifier Block: Each universal quantifier and its associated variable is constructed in G as a universal quantifier block. There is only one way to pebble the universal quantifier block. The sequence of pebbling moves used to pebble the universal quantifier block is shown in Fig. 2.

Existential Quantifier Block: Each existential quantifier and its associated variable is constructed in G as an existential quantifier block. There are two ways to pebble the existential quantifier block depending on whether the associated variable is set to true or false. The sequence of pebbling moves used to pebble the existential quantifier block is shown in Figure 3.

Clause Gadget: A clause gadget is created in G for each clause in F. The clause gadget consists of a pyramid that is connected to each literal in the clause. It was shown in [GLT79] that a pyramid needs as many pebbles as its height to pebble the apex. The clause gadget is shown in Figure 4.

The entire construction using the gadgets described above can be seen in Figure 5.

The key theorem that [GLT79] proves that shows the PSPACE-completeness of the standard pebble game is Theorem 3.

Theorem 3 (Standard Pebble Game is PSPACE-hard [GLT79])**.** *The quantified Boolean formula $B = Q_1 x_1 \cdots Q_u x_u F$ is true if and only if vertex q_1 in graph G (constructed as in e.g. Figure 5) is pebbled with $\mathsf{Peb}(G) = 3m + 3$ pebbles.*

Key Proof Ideas from [GLT79]: We describe the key proof ideas used in [GLT79] to prove the standard pebble game PSPACE-complete that we would need to modify in order for our gap reduction described in Section 3.2 to produce the desired inapproximability result. Using $3m + 3$ pebbles, the constructed graph is pebbled using the following series of steps. The parts that need to be modified in our proof of inapproximability are emphasized below:

1. Given u quantifiers and their associated variables, each of the variables and pyramids in the quantifier blocks are pebbled using s_i, $s_i - 1$, ***and*** $s_i - 2$ ***pebbles where*** $s_1 = 3m+3$. Therefore, before the clauses are pebbled, ***each of the quantifier blocks contains*** 3 ***pebbles***.

2. The clauses are pebbled ***using the additional*** 3 ***pebbles*** in order of construction in the graph.

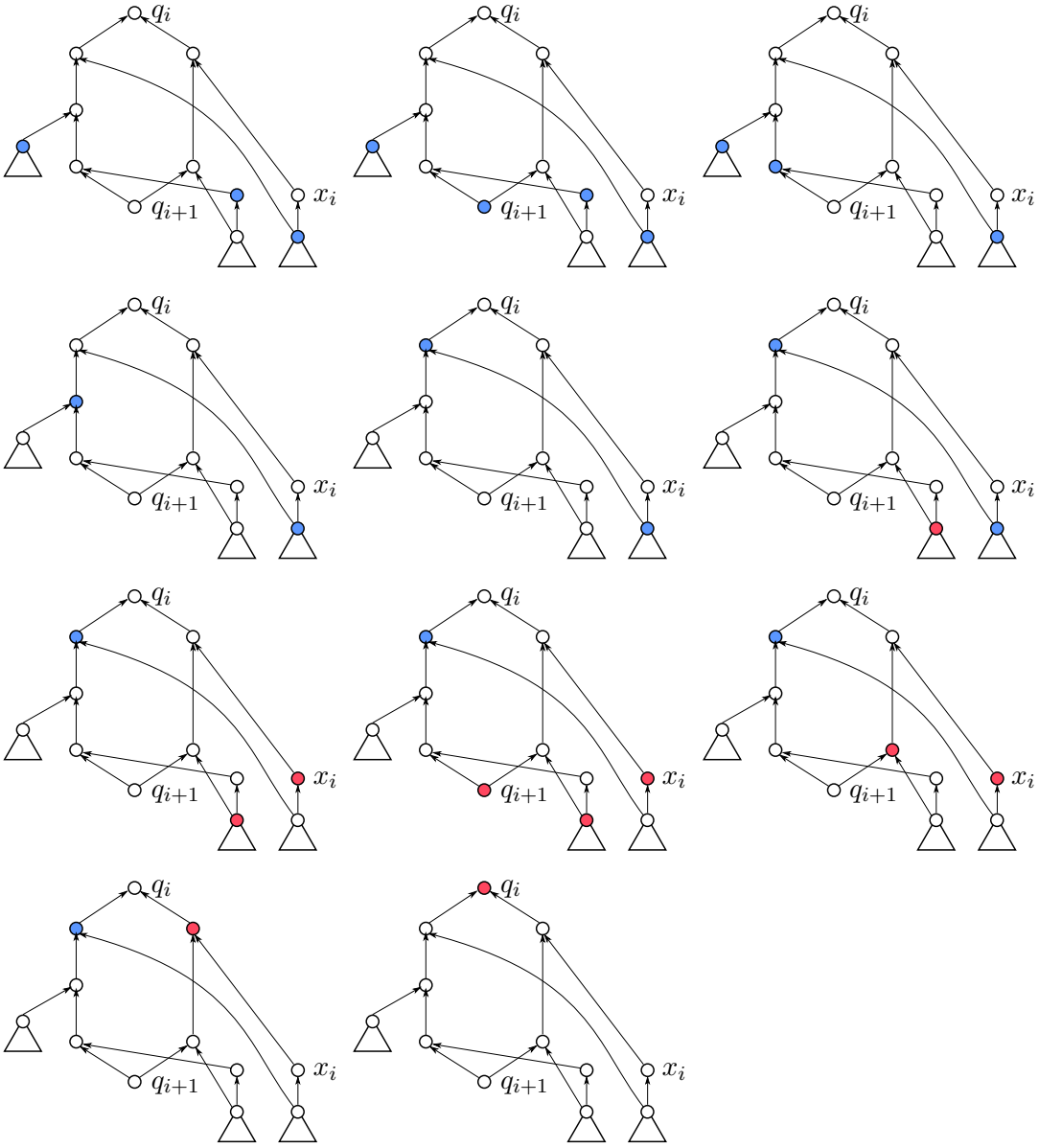

Figure 2: Universal quantifier block and its associated variable. There is only one way to pebble each universal quantifier block. The sequence of pebbling moves used to pebble this gadget is shown above. The change in color of the pebbles signify when all the clauses are repebbled when the universal quantifier is reset. Figure recreated from [GLT79].

3. After all clauses have been pebbled, the remaining portions of the quantifier blocks are pebbled. All pebbles are removed from a quantifier block once that quantifier block has been pebbled. However, most quantifier blocks are pebbled more than once (some potentially 2^v times, see below).

    - If the quantifier, Q_i, is an universal quantifier, then it can only be ***pebbled in one way using*** s_{i-2} ***additional pebbles*** depicted in Figure 2. The change in pebble color

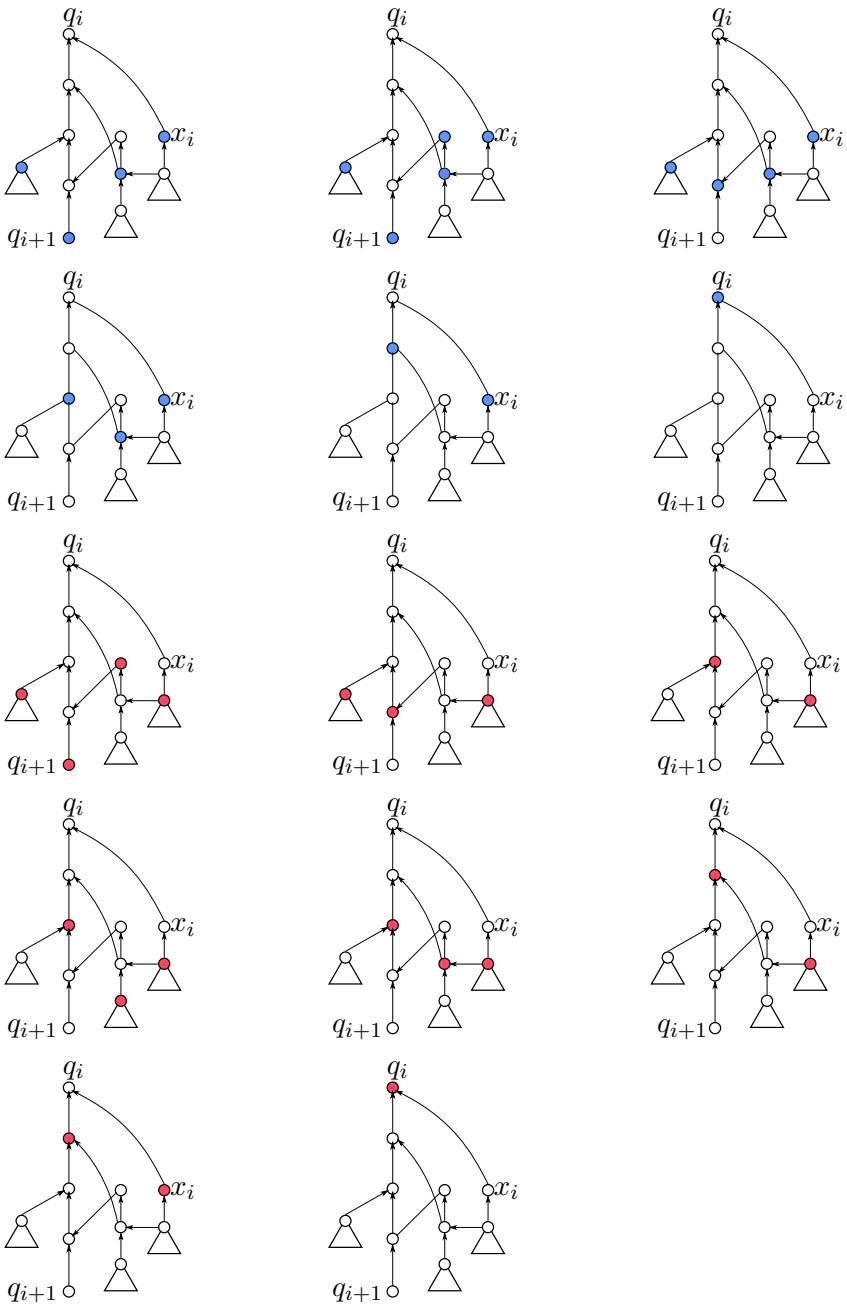

Figure 3: Existential quantifier block and its associated variable. There are two sequences of pebbling moves that result in q_i being pebbled. Blue Pebbles: The sequence of pebbling moves that pebbles q_i when the variable is set to True. Red Pebbles: The sequence of pebbling moves that pebbles q_i when the variable is set to False. Figure recreated from [GLT79].

in Figure 2 indicates when all the clauses and all quantifier blocks below it are repebbled in order to obtain a pebble on q_{i+1}. This inherently means that each clause is checked for satisfiability for each setting of a universal gadget (i.e. setting x_i to both False and

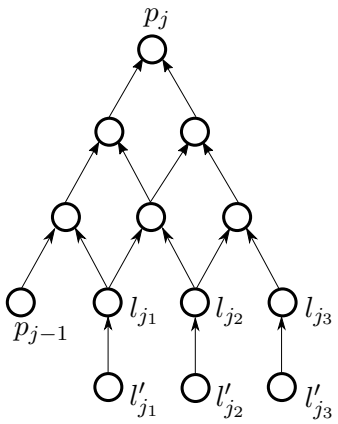

Figure 4: Clause gadget. The clause gadget is pebbled by first pebbling the base of the pyramid using at most 3 extra pebbles. Figure recreated from [GLT79].

True).

- If the quantifier, Q_i, is an existential quantifier, ***it can be pebbled in one of two ways using*** 2 ***or*** s_{i-2} ***additional pebbles*** depicted in Figure 3 depending on whether x_i is set to True or False.

4. Once the last quantifier gadget is pebbled, so is q_1.

To see more specific details of the original proof of the PSPACE-completeness of standard pebbling (including the proof of Theorem 3), please refer to [GLT79].

3.2 Inapproximability to $n^{1/3-\varepsilon}$ additive factor for any $\varepsilon > 0$

We now prove our main result. For our reduction we modify all of the aforementioned gadgets in Section 3.1–variable gadgets, clause gadgets, and quantifier blocks.

3.2.1 Important Subgraphs

Before we dive into the details of our construction, we first mention two subgraphs and the properties they exhibit.

The first graph is the *pyramid graph* (shown in Figure 4 with height 4) Π_h with height h, which requires a number of pebbles that is equal to h to pebble [GLT79]. Therefore, in order to pebble the apex of such a graph, at least h pebbles must be available. As in [GLT79], we depict such pyramid graphs by a triangle with a number indicating the height (hence number of pebbles) needed to pebble the pyramid (see Figure 5 for the triangle symbolism).

We make use of the following definition and lemma (restated and adapted) from [GLT79] in our proofs:

Definition 3 (Frugal Strategy [GLT79])**.** *A pebbling strategy, $\mathcal{P}$, is frugal if the following are true:*

1. *Suppose vertex $v \in G$ is pebbled for the first time at time t'. Then, for all times, $t > t'$, some path from v to q_1 (the only target node) contains a pebble.*

2. *At all times after v is pebbled for the last time, all paths from v to q_1 contain a pebble.*

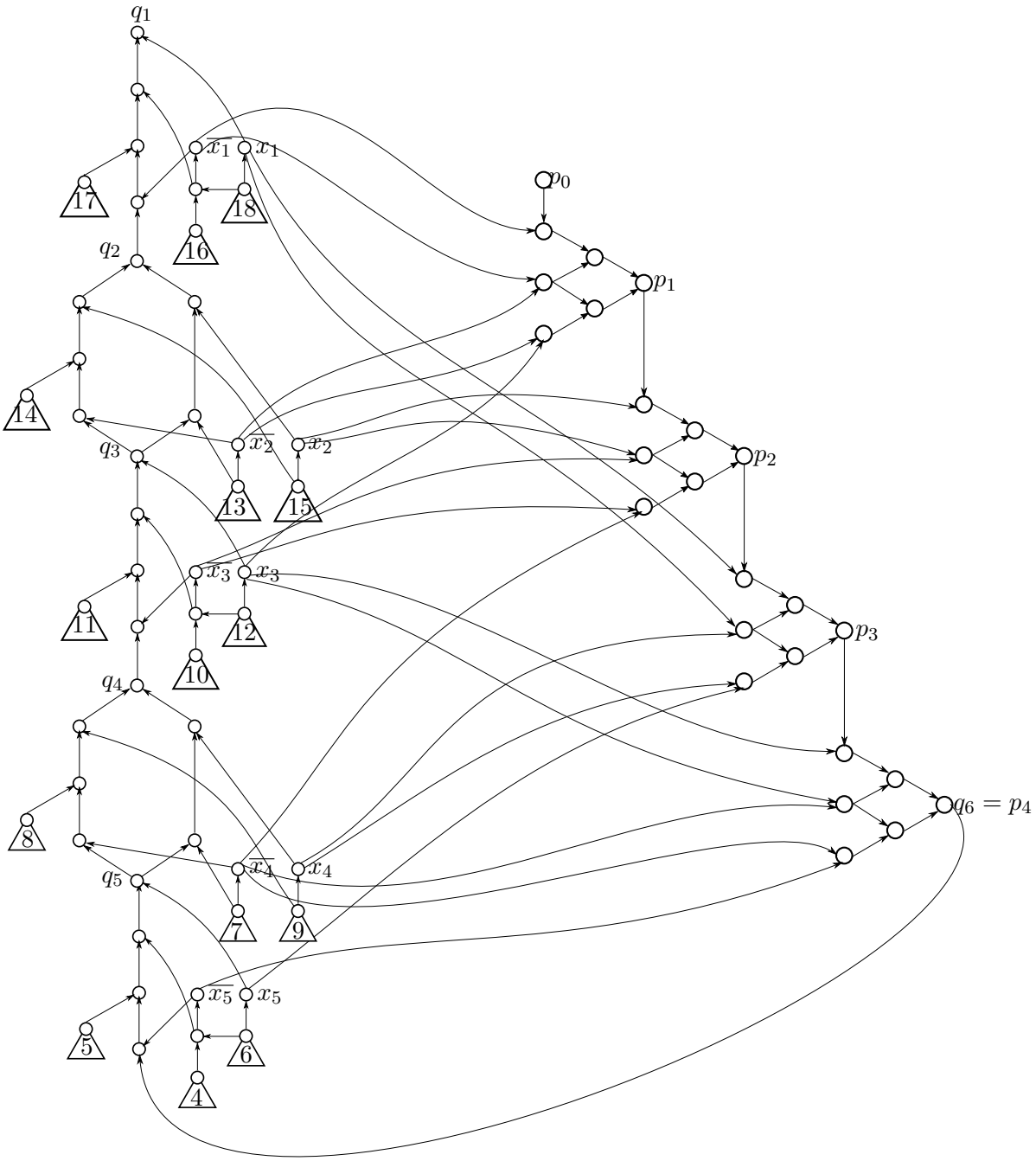

Figure 5: Entire construction of G for $B = \exists x_1 \forall x_2 \exists x_3 \forall x_4 \exists x_5 (\overline{x_1} \vee \overline{x_2} \vee x_3) \wedge (x_2 \vee \overline{x_3} \vee \overline{x_4}) \wedge (x_1 \vee x_4 \vee x_5) \wedge (x_3 \vee \overline{x_4} \vee \overline{x_5})$. This DAG requires 18 pebbles to pebble, $\mathsf{Peb}(G) = 18$, given $T = \{q_1\}$ if and only if B is satisfiable. Figure recreated from [GLT79].

3. *The number of pebble placements on any vertex $v \in G$ where $v \neq q_1$, is bounded by the number of pebble placements on the successors of v.*

Lemma 1 (Normal Pebbling Strategy (adapted from [GLT79]))**.** *If the target vertex is not inside a*

pyramid Π_h *and each of the vertices in the bottom level of the pyramid has at most one predecessor each, then any pebbling strategy can be transformed into a* normal *pebbling strategy without increasing the number of pebbles used. A normal pebbling strategy is one that is frugal and after the first pebble is placed on any pyramid* Π_h *no placements of pebbles occurs outside* Π_h *until the apex of* Π_h *is pebbled and all other pebbles are removed from* Π_h*. Furthermore, no other placement of pebbles occur on* Π_h *until after the pebble on the apex of* Π_h *is removed.*

Note that we can tranform any pyramid that does not fit the requirements of Lemma 1 (i.e. bottom level contains nodes with 2 predecessors) to one that does satisfy the requirments by creating a single predecessor for each node in the bottom level and connecting the original predecessors to this single predecessor.

The other important subgraph we use is the *road graph* (Figure 6), R_w with width w, which requires a number of pebbles that is the width of the graph to pebble *any* of the outputs [EBL79, Nor15]. Therefore, we state as an immediately corollary of the provided proofs:

Corollary 1. *(Road Graph Pebbling) To pebble* $O \subseteq \{o_1, \ldots, o_w\}$ *of the outputs of* R_w*, with a valid strategy,* $\mathcal{P} = [P_0, \ldots, P_\tau]$ *where* $P_\tau = O$*, requires* $w + |O| - 1$ *pebbles.*

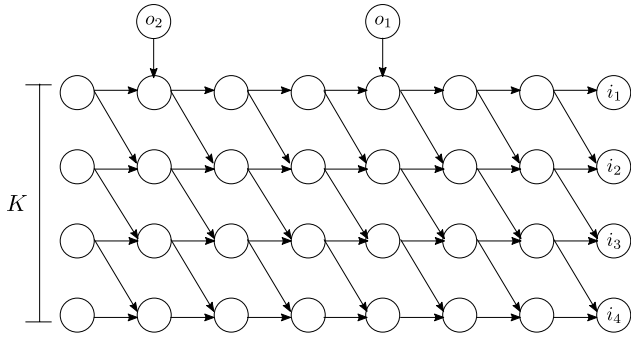

Figure 6: Road graph gadget. Here, in this example, a minimum of 5 pebbles are necessary to pebble o_1 and o_2. 4 pebbles must be used to pebble i_1, i_2, i_3, and i_4 and one more pebble is necessary to pebble o_1 since the four pebbles used to pebble i_1, i_2, i_3 and i_4 must remain on the road graph in order to pebble o_2. K is the width of this road graph gadget. In this example, $K = 4$.

We define a *regular* pebbling strategy for road graphs similarly to the *normal* pebbling strategy for pyramids.

Lemma 2 (Regular Pebbling Strategy)**.** *If each input,* $i_j \in \{i_1, \ldots, i_w\}$*, to a road graph has at most* 1 *predecessor, any pebbling strategy that pebbles a set of desired outputs,* $O \subseteq \{o_1, \ldots, o_w\}$*, at the same time can be transformed into a* regular *pebbling strategy without increasing the number of pebbles used. A regular pebbling strategy is one that is frugal and after the first pebble is placed on any road graph,* R_w*, no placements of pebbles occurs outside* R_w *until the set of desired outputs of* R_w *all contain pebbles and all other pebbles are removed from* R_w*.*

Proof. Consider any pebble strategy, $\mathcal{P}$, that uses s pebbles. We create a regular strategy, $\mathcal{P}'$, that uses at most s pebbles. To create $\mathcal{P}'$, we first delete all unnecessary pebble placements from $\mathcal{P}$, resulting in a frugal strategy that has no unnecessary placements and does not use more than s pebbles. Suppose at time, t_1, a pebble is placed on a road graph, R_w. Let $[t_0, t_2]$ be the largest

time interval containing t_1 such that R_w is never pebble-free during $[t_0, t_2]$. Let μ be the maximum number of pebbles on R_w or any of the ancestors of nodes in R_w during the time interval $[t_0, t_2]$.

By our definition of the frugality of $\mathcal{P}'$ and the fact that the target vertices are not inside R_w, the only pebbles that are on R_w at time t_2 are on the nodes in set O. Since at time $t_0 - 1$ no pebbles are on R_w, there must be a time $t_3 \in [t_0, t_2]$ at which $w + |O| - 1 \leq \mu$ pebbles are on R_w by Corollary 1. Furthermore, for each input, $i_j \in \{i_1, \ldots, i_w\}$, there exists a time $t_{i_j} \in [0, t_2]$ where the predecessor of i_j is pebbled. We modify strategy, $\mathcal{P}$, in the following way to transform it into a regular strategy, $\mathcal{P}'$. Let $\mathsf{pred}(i_j)$ be the set of predecessors of i_j

1. Delete all pebbling placements, removals, and slides on R_w in $[t_0, t_2]$.
2. Delete all pebble placements, removals, and slides on $\mathsf{pred}(i_j)$ and on all ancestors of $\mathsf{pred}(i_j)$ for all $i_j \in \{i_1, \ldots, i_w\}$ in $[t_0, t_2]$.
3. At time t_0, insert a continuous sequence of moves (placements, slides, and removals deleted from 2) that pebbles $\mathsf{pred}(i_j)$ (if it is not pebbled) including any pebble placements on the ancestors of $\mathsf{pred}(i_j)$.
4. At time t_3, insert a continuous sequence of moves that pebbles O using $w + |O| - 1$ pebbles that pebbles all nodes in O and removes all pebbles on $\mathsf{pred}(i_j)$. Then, insert a continuous sequence of moves that removes all pebbles on R_w except for the pebbles on O.

We now prove that the above is a valid strategy that uses at most μ pebbles to pebble R_w in time $[t_0, t_2]$. Steps 1-2 do not increase the pebble count. At time $t_0 - 1$, at most w $\mathsf{pred}(i_j)$ nodes are pebbled using strategy $\mathcal{P}$. Each of these pebbles is moved onto R_w at some time $t_{i_j} \in [t_0, t_2]$ using $\mathcal{P}$. Any pebble originally on $\mathsf{pred}(i_j)$ during the time frame $[t_0, t_2]$ does not get removed from R_w until after all nodes in O are pebbled using $\mathcal{P}$ by construction of the road graph and since $\mathcal{P}$ is a frugal strategy. Therefore, steps 2-3 do not increase the pebble count using $\mathcal{P}'$ since after $\mathsf{pred}(i_j)$ is pebbled, the pebble remains on the graph. Step 3 uses at most $\mu - w - |O| + 1$ additional pebbles to pebble the ancestors of $\mathsf{pred}(i_j)$. Step 4 uses $w + |O| - 1 \leq \mu$ pebbles. □

We immediately obtain the following corollary from Lemmas 1 and 2:

Corollary 2. *Any pebbling strategy, $\mathcal{P}$, can be transformed into a pebbling strategy, $\mathcal{P}'$, that is normal and regular if no target vertices lie inside a pyramid or road graph and each input node to either the pyramid or road graph has at most one predecessor.*

3.2.2 Modified Graph Constructions

We first describe the changes we made to each of the gadgets used in the PSPACE-completeness proof presented by [GLT79] and then prove our inapproximability result using these gadgets in Section 3.2.4. Given a QBF instance, $B = Q_1 x_1 \cdots Q_u x_u F$, with c clauses, we create the following gadgets:

Variable Nodes: Suppose we now replace all paths in variable nodes in the proof provided by [GLT79] (see Figure 1) with road graphs each of width K. The modified variable gadgets are shown in Figure 7. Each variable gadget as in the original proof by [GLT79] has 3 possible configurations which are also shown in Figure 7.

Quantifier Blocks: Each universal and existential quantifier block is also modified to account for the new variable gadgets. See Figure 8 and Figure 9 which depict the new quantifier blocks that

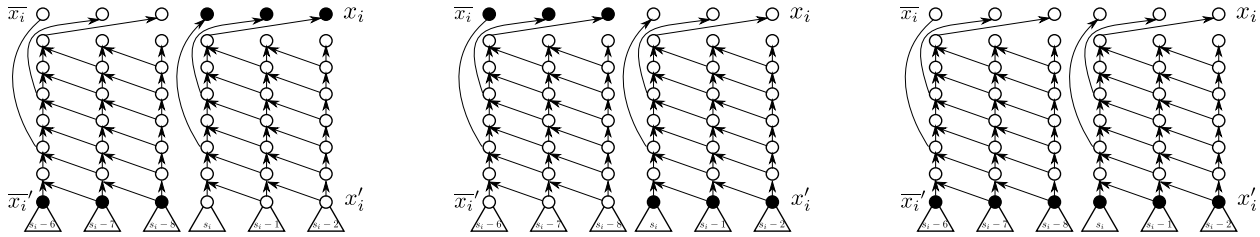

Figure 7: Modified variable gadget with 3 possible configurations using the road graph previously described. Here $K = 3$. a) x_i is True. b) $\overline{x_i}$ is True. c) Double false.

use the new variable gadgets. Note that instead of each quantifier block requiring 3 total pebbles (as in the proof sketch described in Section 3.1), each gadget requires $3K$ pebbles to remain before the clauses are pebbled. The basic idea is to expand all nodes $a_i, b_i, c_i ...etc.$ into a path of length K with interconnections to account for each of the K copies of x_i and each of K copies of $\overline{x_i}$. Each $s_i = s_{i-1} - 3K$ and $s_1 = 3Ku + 4K + 1$.

Clause Gadgets: Each clause gadget is modified to be a pyramid of height $3K + 1$ where the bottom layer is connected to nodes from a combinations of different literals (as its ancestors). Therefore, for a given clause (l_i, l_j, l_k), K nodes are connected to l_i, l_j, l_k, K nodes to l_i and l_k, and K nodes to l_j and l_k. See Figure 10 for an example of the modified clause gadget.

3.2.3 Full Contruction

Using the modified gadgets described in Section 3.2.2, we create the full construction of a graph G from an instance of QBF B using a similar construction to that shown in Figure 5 except with the following key modifications:

1. $s_1 = 3Ku + 4K + 1$ and $s_i = s_{i-1} - 3K$ for all $1 < i \leq u$.
2. In the figures of the modified gadgets (Figs. 10, 9, 8, and 11), all vertices with indegree 3 are replaced with pyramids of height 3.
3. Each q_i and p_j are the apex of pyramids of height K. Let $\{q_i^1, \ldots, q_i^K\}$ and $\{p_j^1, \ldots, p_j^K\}$ be the nodes on the bottom level of each pyramid, respectively. Each q_i is connected via outgoing directed edges to K other nodes. We refer to these nodes as $\{q_i^{1'}, \ldots, q_i^{K'}\}$. There exist edges $(q_i, q_i^{l'})$ for all $1 \leq l \leq K$, (p_j, p_{j+1}), $(q_i^l, q_i^{l'})$ for all $1 \leq l \leq K$, and (p_j^l, p_{j+1}^l) for all $1 \leq l \leq K$.
4. The first clause gadget in the topological sort order of clause gadgets is a duplicate of the first clause. In other words, the first two clauses gadgets are the same in the topological sort order of the clause gadgets.
5. The target node q_1 is the apex of a pyramid of height K.
6. The clause gadget has height $3K + 1 - K = 2K + 1$ where the top K nodes of the clause pyramid are connected to the corresponding $\{p_i^1, \ldots, p_i^K\}$.

3.2.4 Proofs of the Construction

We construct a graph G using the construction described above in Section 3.2.3 for any given QBF instance, $B = Q_1 x_1 \cdots Q_u x_u F$. In short, the proof relies on the fact that each quantifier block requires $2K$ pebbles to set the corresponding variable to true, false, or double false (i.e. the corresponding literals to true or false). An additional K pebbles need to remain on each quantifier in order to be able to repebble quantifiers when checking for universal variables' satisfaction. Furthermore, a clause would consist of modified pyramids of height $3K + 1$ connected to pairs of nodes from different literals. Following the proof in [GLT79], the quantifier blocks are pebbled first with $3Ku$ pebbles remaining on the quantifier blocks. Then, the clauses are pebbled with $3K + 1$ pebbles.

In the full construction, we include a duplicate copy of the first clause since the first clause can always be pebbled with $4K + 1$ pebbles regardless of whether or not it is satisfiable by the variable assignments. The variables, quantifiers blocks, and clauses are otherwise connected similarly to the construction presented in [GLT79].

If B is satisfiable, then clauses can be pebbled with $4K + 1$ pebbles. Otherwise, $5K$ pebbles are needed to pebble one or more unsatisfied clauses in G, resulting in a gap of $K - 1$ pebbles between when B is satisfiable and unsatisfiable. Thus, if given an approximation algorithm that estimates the number of pebbles needed within an additive $K - 1$, we can distinguish between the case when B is satisfiable (at most $3Km + 4K + 1$ pebbles are needed) and the case when B is unsatisfiable (when $3Km + 5K$ pebbles are needed).

In this construction, K can be any polynomial function of u and c where u is the number of variables in B and c is the number of clauses (in other words, $K = u^a c^b$ for any constants a and b). The total minimum number of pebbles necessary is $O(Ku)$ and the total number of nodes in the constructed graph is $O(K^3(u^3 + c))$.

Suppose time t_1 is the time when the first pebble is placed on the first clause in the topological order of the clause gadgets. Let t_0 be some time where $t_0 < t_1$ when p_0 is first pebbled. Let t_2 be the time when q_1 is pebbled. We first prove the following lemma.

Lemma 3. *At least K pebbles remain on the paths from p_0 to q_1 during the time frame $[t_0, t_2]$ given a normal and regular strategy, $\mathcal{P}'$, that can pebble the graph using any arbitrary number of pebbles.*

Proof. For any strategy, $\mathcal{P}$, that pebbles the graph, by Corollary 2 we show that we can transform it into a normal and regular strategy, $\mathcal{P}'$, where K pebbles remain on the paths from p_0 to q_1 during the timeframe $[t_0, t_2]$.

At time t_1, in order to pebble the first clause, p_0 must be completely pebbled requiring at least K pebbles using strategy $\mathcal{P}'$. By the normality and regularity of $\mathcal{P}'$, K pebbles remain on $\{p_i^1, \ldots, p_i^K\}$ for all $i \in [0, c]$. If $k \leq K$ pebbles are ever removed from $\{p_i^1, \ldots, p_i^K\}$, then one of three cases can occur:

1. If less than K pebbles remain on any of the paths from p_0 to p_c in total, then all clause gadgets $C_{i'}$ for all $i' \leq i$ must be repebbled.

2. If less than K pebbles remain on $\{p_{i'}^1, \ldots, p_{i'}^K\}$ in total for all $i' \leq i$ and some $p_{i'}$ where $i' \leq i$ can be repebbled, then K pebbles must remain on the top of the pyramids for clauses $C_{i'}$ where $i' \leq i$.

3. If not all $\{p_{i'}^1, \ldots, p_{i'}^K\}$ are repebbled and less than K pebbles remain in total on the top of the pyramids for clauses $C_{i'}$ where $i' \leq i$, then a total of K pebbles remain on $\{p_{i'}^1, \ldots, p_{i'}^K\}$ for all $i' \leq i$.

Case 1 violates the normality and regularity of $\mathcal{P}'$. Cases 2 and 3 both still maintain K pebbles on the paths from p_0 to q_1, thus not violating the lemma.

Now we show that at the conclusion of pebbling the clauses, K pebbles remain on $\{p_m^1, \ldots, p_m^K\} = \{q_{u+1}^1, \ldots, q_{u+1}^K\} = \{q_{u+1}^{1'}, \ldots, q_{u+1}^{K'}\}$. Suppose less than K pebbles remain on $\{p_m^1, \ldots, p_m^K\}$, then Q_u trivially cannot be pebbled and one of the three cases above must be true.

We now argue that K pebbles remain on either $\{q_{i+1}^1, \ldots, q_{i+1}^K\}$, $\{c_i^1, \ldots, c_i^K\}$, $\{b_i^1, \ldots, b_i^K\}$, $\{a_i^1, \ldots, a_i^K\}$, $\{g_i^1, \ldots, g_i^K\}$, or $\{f_i^1, \ldots, f_i^K\}$ during the pebbling of a universal quantifier block or on $\{q_{i+1}^1, \ldots, q_{i+1}^K\}$, $\{a_i^1, \ldots, a_i^K\}$, $\{b_i^1, \ldots, b_i^K\}$, $\{c_i^1, \ldots, c_i^K\}$, or $\{e_i^1, \ldots, e_i^K\}$ during the pebbling of an existential quantifier block.

When pebbling the first quantifier block, we consider two cases:

1. Universal quantifier block: K pebbles must be moved from $\{q_{u+1}^{1'}, \ldots, q_{u+1}^{K'}\}$ to $\{c_i^1, \ldots, c_i^K\}$. When pebbling a $False$ assignment to the variable, the K pebbles must remain on $\{c_i^1, \ldots, c_i^K\}$ until the last of the nodes in the topological sort order of $\{c_i^1, \ldots, c_i^K\}$ is pebbled since $\{b_i^1, \ldots, b_i^K\}$ cannot be pebbled until the last of the $\{c_i^1, \ldots, c_i^K\}$ is pebbled and one of the predecessors of b_i^1 is c_i^1. By the same argument, K pebbles are moved onto $\{b_i^1, \ldots, b_i^K\}$ then $\{a_i^1, \ldots, a_i^K\}$ and remain on either $\{a_i^1, \ldots, a_i^K\}$ or $\{b_i^1, \ldots, b_i^K\}$ until $\{q_u^1, \ldots, q_u^K\}$ is pebbled. When pebbling a $True$ assignment to the variable, we follow the same logic that K pebbles must be moved onto $\{g_i^1, \ldots, g_i^K\}$ and then $\{f_i^1, \ldots, f_i^K\}$ until K pebbles are used to pebble $\{q_u^1, \ldots, q_u^K\}$.

2. Existential quantifier block: K pebbles must be moved from $\{q_{u+1}^{1'}, \ldots, q_{u+1}^{K'}\}$ to $\{e_i^1, \ldots, e_i^K\}$. No pebbles can be removed from $\{e_i^1, \ldots, e_i^K\}$ until the final element in the topological sort order of $\{e_i^1, \ldots, e_i^K\}$ is pebbled since to pebble the first element of $\{c_i^1, \ldots, c_i^K\}$, we need to pebble the last element of $\{e_i^1, \ldots, e_i^K\}$. Then, K pebbles are transferred from $\{e_i^1, \ldots, e_i^K\}$ to $\{c_i^1, \ldots, c_i^K\}$ to $\{b_i^1, \ldots, b_i^K\}$, to $\{a_i^1, \ldots, a_i^K\}$, and finally to $\{q_u^1, \ldots, q_u^K\}$.

The rest of the proof for $\{q_i^1, \ldots, q_i^K\}$ where $1 \leq i < u$ follows easily from induction by using the base case provided above. □

We then prove that the number of pebbles needed to pebble each quantifier block is $3K$ and $3K$ pebbles remain on the quantifier blocks throughout the pebbling of the clauses.

Lemma 4. *Let N_i be the configuration such that some number of pebbles are on the first $i-1$ quantifier blocks and the i-th quantifier block (i.e. $\{q_{i+1}^1, \ldots, q_{i+1}^K\}$ contains K pebbles and $\{q_i^1, \ldots, q_i^K\}$ does not yet contain K pebbles) is being pebbled. Therefore, N_{u+1} is the configuration when some number of pebbles are on all u quantifier blocks and the first clause gadget is being pebbled. There does not exist a normal and regular strategy, $\mathcal{P}$, that uses less than $3Ku + 5K$ pebbles that can pebble our reduction construction, G, such that N_i contains less than $s - s_i$ pebbles on the first $i-1$ quantifier blocks when the i-th quantifier block or when the first clause is being pebbled.*

Proof. Let Q_i be the i-th quantifier block. If N_{u+1} contains less than $s - s_{u+1} = 3Ku$ pebbles, then there exists a quantifier block, Q_j, that contains less than $3K$ pebbles when the clauses are being pebbled. Let Q_j be the first quantifier block that is missing at least one pebble (i.e. N_j contains

$s - s_j$ pebbles). If less than $3K$ pebbles are on the Q_j block then two possible scenarios could occur. Either:

1. Some literal configuration contains less than K pebbles.
2. d_j^l is not pebbled for some $1 \leq l \leq K$.

Let t_c be the time when the first clause is pebbled. Suppose on the contrary that less than $3K$ pebbles were placed on each quantifier block at time $t_c - 1$. Let the unpebbled vertex be v and let v be part of quantifier block Q_j. If v is part of a literal in a clause, then v must be pebbled in order to pebble the clauses. Let time t' be the time when a clause containing v is pebbled. Then, v must be pebbled at t'. To pebble v at time t', at least $s' \geq s_j - 3K + 1$ pebbles must be removed from the graph to pebble the literal.

Without loss of generality, we assume that the apex of the pyramid with height $s_i - 3K + 1$ is missing a pebble at vertex v. Note that our argument applies for any of the pyramids in the gadget that is missing a pebble. For any other pyramid that is missing a pebble in Q_j, we need only consider whether pebbles remain on Q_j itself.

Since our strategy $\mathcal{P}$ uses less than $3Ku + 5K$ pebbles in total to pebble G, there are two ways to obtain the necessary s' pebbles:

1. Remove s' pebbles from clauses and quantifier blocks $Q_{j'}$ where $j' > j$. Then, we must remove all pebbles except for at most $K - 1$ pebbles that can remain on the clause gadgets or the quantifier blocks $Q_{j'}$. Thus we know that at most $K - 1$ pebbles total can be on the paths from p_0 to p_c. By Lemma 3, this contradicts the normality and regularity of $\mathcal{P}$.
2. Remove pebbles from $Q_{j'}$ where $j' < j$. Then, our argument for Case 1 applies to $Q_{j'}$ and so on until no quantifier block with lower order number contains less than $3K$ pebbles.

Thus if $N_{u+1} < 3Ku$, then the normality and regularity of $\mathcal{P}$ is violated.

If d_j^i is not pebbled for some $1 \leq i \leq K$, then, when d_j^i must be pebbled, at least $s_i - 2K + 1$ pebbles must be removed from the graph in order to pebble d_j^i. Lemma 3 is again violated as at most $K - 1$ pebbles total can remain on the paths from p_0 to q_1. □

Next we prove that provided $3Km$ pebbles stay on the quantifier blocks, each unsatisfied clause requires $5K$ pebbles.

Lemma 5. *Given a clause gadget, C_i, where $i > 1$ (since the first clause is a duplicate), its corresponding clause, c_i is true if and only if C_i can be pebbled with $4K + 1$ pebbles (including the K pebbles on p_{i-1}) and no pebbles are added, removed, or slid on the literals attached to the clause gadget. Furthermore, if all literals in C_i are set in the false configuration, then at least $5K$ pebbles are necessary to pebble the clause (including the K pebbles on p_{i-1}).*

Proof. We first prove that if c_i is true, then C_i can be pebbled using $4K + 1$ pebbles. Given any valid strategy $\mathcal{P}$, we first note that we can transform $\mathcal{P}$ into a regular strategy where the pyramid in Figure 10 with apex p_j can be pebbled using $4K + 1$ pebbles. Because c_i is true, at least one of the literals in c_i must be true. Therefore, one of the literals connected to the bottom of the pyramid in C_i must be in the true configuration. $4K$ pebbles can then be used to pebble the other two literals, $\{p_{j-1}^1, \ldots, p_{j-1}^K\}$, and $3K + 1$ pebbles can then be used to pebble the clause pyramid. One can check a small number of cases to see that this is true. For example, in Fig. 10, we have the following cases:

1. Suppose that $x_i = True$, then using $3K$ pebbles, we pebble $\{x_l^1, \ldots, x_l^K\}$ and $\{x_r^1, \ldots, x_r^K\}$, leaving $2K$ pebbles. We pebble the nodes in the bottom layer of the pyramid that have $\{x_l^1, \ldots, x_l^K\}$ and $\{x_r^1, \ldots, x_r^K\}$ as predecessors with the K remaining pebbles. Then, we move the K pebbles on $\{x_l^1, \ldots, x_l^K\}$ to $\{y_i^1, \ldots, y_i^K\}$ (assuming by Lemma 3 that $\{p_{j-1}^1, \ldots, p_{j-1}^K\}$ contains K pebbles) and to the bottom level of the clause pyramid. Finally, we move the K pebbles from $\{x_r^1, \ldots, x_r^K\}$ to the bottom level of the clause pyramid.

2. Suppose that $x_l = True$, then using $3K$ pebbles, we pebble $\{x_i^1, \ldots, x_i^K\}$ and $\{x_r^1, \ldots, x_r^K\}$. Use remaining K pebbles to pebble nodes on bottom layer of clause pyramid that have $\{x_i^1, \ldots, x_i^K\}$ and $\{x_r^1, \ldots, x_r^K\}$ as predecessors. Move K pebbles from $\{x_i^1, \ldots, x_i^K\}$ to $\{y_i^1, \ldots, y_i^K\}$ and then to bottom layer of clause pyramid. Move K pebbles from $\{x_r^1, \ldots, x_r^K\}$ to bottom layer.

3. Suppose that $x_r = True$, then using $3K$ pebbles, we pebble $\{x_i^1, \ldots, x_i^K\}$ and $\{x_l^1, \ldots, x_l^K\}$. Use remaining K pebbles to pebble $\{y_i^1, \ldots, y_i^K\}$ and then bottom layer of clause pyramid. Move K pebbles from $\{x_i^1, \ldots, x_i^K\}$ to bottom layer. Move K pebbles from $\{x_l^1, \ldots, x_l^K\}$ to bottom layer.

Now we prove the more difficult direction that if C_i can be pebbled using $4K+1$ pebbles, then c_i is true. Each variable gadget requires K pebbles to pebble each output. There exists a time t_1 when $3K+1$ pebbles are on the clause pyramid. By regularity of the pyramid, a total of $3K+1$ pebbles must be on the predecessors of the bottom level of the pyramid before any pebbles are placed on the pyramid. However, at most $2K+2$ pebbles can be placed on the pyramid if none of the literals are set in the true configuration. K pebbles must be on p_{i-1} or the frugality of the strategy is violated as proven in Lemma 3. The only entry point into the pyramid is through the literals. However, every output requires K pebbles to pebble. For every pebble that enters the pyramid, at least K pebbles must be free to pebble the road graph. However, if none of the literals are true, then to place the p-th pebble where $2K+3 \leq p \leq 3K$ on the pyramid requires one additional pebble each which we remove from other parts of the graph, a contradiction to our assumption.

Following the argument presented above, to pebble each output of a literal requires K pebbles. Therefore, to move the $3K+1$ pebbles onto the clause pyramid, we need $K-1$ additional pebbles, resulting in $5K$ pebbles. □

Given the previous proofs, we now prove the following key lemmas:

Lemma 6. *Given G which is constructed from the provided QBF instance, $B = Q_1 x_1 \cdots Q_u x_u F$, using our modified reduction in Section 3.2.2, B is satisfiable if and only if $\mathsf{Peb}(G) \leq 3Ku + 4K + 1$.*

Proof. We first prove that if B is satisfiable, then the graph can be pebbled with $3Ku + 4K + 1$ pebbles. We prove this via induction, similar to the proof given in [GLT79]. Let $s = 3Ku + 4K + 1$ and s_i be defined as in Section 3.2.3. For $1 \leq i \leq u+1$, we define N_i to be the set of configurations fixing truth values to the first $i-1$ variable nodes. An arrangement of exactly $s - s_i$ pebbles on G is in N_i if and only if, for $1 \leq j < i$, the following two conditions hold:

1. If $Q_j = \forall$, then exactly $3K$ pebbles are on the j-th quantifier block, on one of the following three sets of vertices:
   (a) $\{d_j^1, \ldots, d_j^K\} \cup \{x_j^1, \ldots, x_j^K\} \cup \{\overline{x_j^{1\prime}}, \ldots, \overline{x_j^{K\prime}}\}$, indicating x_j is $True$;
   (b) $\{d_j^1, \ldots, d_j^K\} \cup \{\overline{x_j^1}, \ldots, \overline{x_j^K}\} \cup \{x_j^{1\prime}, \ldots, x_j^{K\prime}\}$, indicating x_j is $False$;

(c) $\{d_j^1, \ldots, d_j^K\} \cup \{x_j^{1\prime}, \ldots, x_j^{K\prime}\} \cup \{\overline{x_j^{1\prime}}, \ldots, \overline{x_j^{K\prime}}\}$, indicating a double $False$.

2. If $Q_j = \exists$, then exactly $3K$ pebbles are on the j-th quantifier block, on one of the following three sets of vertices:

   (a) $\{d_j^1, \ldots, d_j^K\} \cup \{x_j^1, \ldots, x_j^K\} \cup \{\overline{x_j^{1\prime}}, \ldots, \overline{x_j^{K\prime}}\}$ indicating x_j is $True$;

   (b) $\{d_j^1, \ldots, d_j^K\} \cup \{\overline{x_j^1}, \ldots, \overline{x_j^K}\} \cup \{x_j^{1\prime}, \ldots, x_j^{K\prime}\}$, indicating x_j is $False$;

   (c) $\{d_j^1, \ldots, d_j^K\} \cup \{x_j^{1\prime}, \ldots, x_j^{K\prime}\} \cup \{\overline{x_j^{1\prime}}, \ldots, \overline{x_j^{K\prime}}\}$, indicating a double $False$.

By our definition, N_1 contains no pebbles on the graph, and N_{u+1} contains all configurations in which a truth assignment has been made to each literal and $4K+1$ pebbles remain to test whether the assignment makes F true.

We now prove the following claim which subsequently also proves that if B is satisfiable then $\mathsf{Peb}(G) \leq 3Ku + 4K + 1$.

Claim 1. *Let $1 \leq i \leq u+1$. Suppose the graph is initially in a configuration N_i. For $1 \leq j < i$, let e_{2j-1} be the truth assignment defined for x_j by that configuration, and let e_{2j} be the truth assignment defined for $\overline{x_j}$. If $Q_i x_i \cdots Q_u x_u F(e_1, e_2, \cdots, e_{2i-3}, e_{2i-2})$ is true, then vertex q_i can be pebbled with additional pebbles without moving any of the $s - s_i$ pebbles initially on the graph.*

Proof. We prove by induction on i from $u+1$ to 1.

Let $i = u+1$ and suppose that the assignment defined by the N_i configuration makes F true. We must show that any vertex $\{q_{u+1}^{1'}, \ldots, q_{u+1}^{K'}\} = \{p_c^1, \ldots, p_c^K\}$ can be pebbled with $s_{u+1} = 4K+1$ pebbles without moving any of the pebbles of the N_{u+1} configuration. We showed in Lemma 5 that this is the case.

Now suppose that the lemma holds for $i+1$ so that the assignment defined by the N_i configuration makes the substituted formula $Q_i x_i \cdots Q_u x_u F(e_1, e_2, \cdots, e_{2i-3}, e_{2i-2})$ true.

To prove that the lemma holds for i, there are two cases we have to consider:

1. Suppose $Q_i = \forall$. Then,

$$Q_{i+1} x_{i+1} \cdots Q_u x_u F(e_1, \cdots, e_{2i-2}, True, False) \text{ and}$$
$$Q_{i+1} x_{i+1} \cdots Q_u x_u F(e_1, \cdots, e_{2i-2}, False, True)$$

   are both true.

   Vertices q_i, $\{q_i^1, \ldots, q_i^K\}$, and $\{q_i^{1'}, \ldots, q_i^{K'}\}$ can be pebbled with s_i pebbles as follows. First, use all s_i pebbles to pebble $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$, leaving K pebbles, one on the apex of each of the pyramids. Then, use the remaining $s_i - K$ pebbles to pebble $\{d_i^1, \ldots, d_i^K\}$ leaving K pebbles, one on each of the d_i's. Finally, with $s_i - 2K$ remaining pebbles, pebble $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$, then move the K pebbles on $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$ to $\{\overline{x_i^1}, \ldots, \overline{x_i^K}\}$. The current configuration is in N_{i+1} representing the variable $x_i = False$. Applying the induction hypothesis, pebble q_{i+1}, $\{q_i^1, \ldots, q_i^K\}$, and $\{q_i^{1'}, \ldots, q_i^{K'}\}$ with the remaining $s_{i+1} = s_i - 3K$ pebbles. Move the pebbles on $\{q_{i+1}^{1'}, \ldots, q_{i+1}^{K'}\}$ to $\{c_i^1, \ldots, c_i^K\}$, $\{b_i^1, \ldots, b_i^K\}$, and $\{a_i^1, \ldots, a_i^K\}$. Move the pebbles on $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$ to $\{x_i^1, \ldots, x_i^K\}$. Leaving pebbles on $\{a_i^1, \ldots, a_i^K\}$ and $\{x_i^1, \ldots, x_i^K\}$, pick up the rest of the pebbles and use the $s_i - 2K$ free pebbles to pebble $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$, leaving

K pebbles there. The current configuration is in N_{i+1}, representing the variable $x_i = True$. Applying the induction hypothesis, pebble $\{q_{i+1}^{1'}, \ldots, q_{i+1}^{K'}\}$ again. Finish by moving the K pebbles on $\{q_{i+1}^{1'}, \ldots, q_{i+1}^{K'}\}$ to $\{g_i^1, \ldots, g_i^K\}$, $\{f_i^1, \ldots, f_i^K\}$, and $\{q_i^1, \ldots, q_i^K\}$.

If $Q_{i+1}x_{i+1} \cdots Q_u x_u F(e_1, \cdots, e_{2j-2}, False, False)$ is true, there is a way to pebble $\{q_i^1, \ldots, q_i^K\}$ that only pebbles $\{q_{i+1}^{1'}, \ldots, q_{i+1}^{K'}\}$ once. First pebble $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$, $\{d_i^1, \ldots, d_i^K\}$, and $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$, which gives a configuration in N_{i+1} representing the literals x_i and $\overline{x_i}$ are both false. Applying the induction hypothesis, pebble $\{q_{i+1}^{1'}, \ldots, q_{i+1}^{K'}\}$. There are now $s_i - 4K \geq 2K$ free pebbles. Place K on $\{\overline{x_i^1}, \ldots, \overline{x_i^K}\}$ and move them to $\{c_i^1, \ldots, c_i^K\}$, $\{b_i^1, \ldots, b_i^K\}$, and $\{a_i^1, \ldots, a_i^K\}$. Move the pebbles on $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$ to $\{g_i^1, \ldots, g_i^K\}$ and finish by moving the pebbles on $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$ to $\{x_i^1, \ldots, x_i^K\}$, $\{f_i^1, \ldots, f_i^K\}$, and $\{q_i^1, \ldots, q_i^K\}$.

2. Suppose $Q_i = \exists$. Then either

$$Q_{i+1}x_{i+1} \cdots Q_u x_u F(e_1, \cdots, e_{2i-2}, True, False) \text{ or}$$
$$Q_{i+1}x_{i+1} \cdots Q_u x_u F(e_1, \cdots, e_{2i-2}, False, True)$$

is true.

Suppose that the former is the case. Vertices $\{q_i^1, \ldots, q_i^K\}$, q_i, and $\{q_i^{1'}, \ldots, q_i^{K'}\}$ can be pebbled with s_i pebbles as follows. First pebble $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$ leaving pebbles there. Then, pebble $\{d_i^1, \ldots, d_i^K\}$ and $\{f_i^1, \ldots, f_i^K\}$, leaving pebbles there. Move the pebbles on $\{f_i^1, \ldots, f_i^K\}$ to $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$ and move the pebbles on $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$ to $\{x_i^1, \ldots, x_i^K\}$. The current configuration is in N_{i+1}, representing variable $x_i = True$. Applying the induction hypothesis, pebble $\{q_{i+1}^{1'}, \ldots, q_{i+1}^{K'}\}$ with the remaining $s_{i+1} = s_i - 3K$ pebbles. There are now $s_i - 4K \geq 2K$ free pebbles. Place K pebbles on $\{\overline{x_i^1}, \ldots, \overline{x_i^K}\}$ and finish by moving the K pebbles on $\{\overline{x_i^1}, \ldots, \overline{x_i^K}\}$ to $\{e_i^1, \ldots, e_i^K\}$, $\{c_i^1, \ldots, c_i^K\}$, $\{b_i^1, \ldots, b_i^K\}$, $\{a_i^1, \ldots, a_i^K\}$, and $\{q_i^1, \ldots, q_i^K\}$.

Alternatively, suppose that $Q_{i+1}x_{i+1} \cdots Q_u x_u F(e_1, \cdots, e_{2i-2}, False, True)$ is true. To pebble $\{q_i^1, \ldots, q_i^K\}$ with s_i pebbles, begin by pebbling $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$, $\{d_i^1, \ldots, d_i^K\}$, and $\{f_i^1, \ldots, f_i^K\}$ in turn, leaving pebbles there. Move the pebbles on $\{f_i^1, \ldots, f_i^K\}$ to $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$ and $\{\overline{x_i^1}, \ldots, \overline{x_i^K}\}$, which gives a configuration in N_{i+1} representing variable $x_i = False$. Applying the induction hypothesis, pebble $\{q_{i+1}^{1'}, \ldots, q_{i+1}^{K'}\}$. Move the pebbles on $\{q_{i+1}^{1'}, \ldots, q_{i+1}^{K'}\}$ to $\{e_i^1, \ldots, e_i^K\}$ and $\{c_i^1, \ldots, c_i^K\}$. Pick up all the pebbles except for those on $\{c_i^1, \ldots, c_i^K\}$ and $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$ and use the $s_i - 2K$ free pebbles to pebble $\{f_i^1, \ldots, f_i^K\}$. Move the pebbles on $\{f_i^1, \ldots, f_i^K\}$ to $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$, then $\{b_i^1, \ldots, b_i^K\}$, and finish by moving the pebbles on $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$ to $\{x_i^1, \ldots, x_i^K\}$, then $\{a_i^1, \ldots, a_i^K\}$, and finally to $\{q_i^1, \ldots, q_i^K\}$.

□

When $i = 1$, the proof of of the above Claim 1 proves that if B is satisfiable then $\mathsf{Peb}(G) \leq 3Ku + 4K + 1$.

Now we prove that if $\mathsf{Peb}(G) \leq 3Km + 4K + 1$, then B is satisfiable. Note that in the subsequent proofs, we assume that we only remove pebbles from quantifiers blocks with higher order number since we proved in Lemma 3 that removing pebbles from quantifier blocks with lower order number violates the normality and regularity of a pebbling strategy. We first prove the following claim which subsequently proves this.

Claim 2. *Let $1 \leq i \leq n+1$. Suppose the graph is initially in a configuration in N_i. For $1 \leq j < i$, let e_{2j-1} be the truth assignment defined for x_i by that configuration, and let e_{2j} be the truth assignment for $\overline{x_j}$. If vertex q_i can be pebbled with s_i additional pebbles without moving any of the $s - s_i$ pebbles initially on the graph, then $Q_i x_i \cdots Q_n x_n(e_1, e_2, \cdots, e_{2i-3}, e_{2i-2})$ is true.*

Proof. We assume that any strategy $\mathcal{P}$ that can pebble G using $3Ku+4K+1$ pebbles is transformed into a normal and regular strategy $\mathcal{P}'$ that pebbles G using $3Ku+4K+1$ pebbles. Again, we prove by induction on i from $u+1$ to 1. In the base case, let $i = u+1$, then by Lemma 4 and Lemma 5, each clause gadget contains at least one literal gadget in the $True$ configuration.

Suppose by induction that the lemma holds for $i+1$, we now prove there is a strategy which pebbles $\{q_i^1, \ldots, q_i^K\}$ with s_i pebbles without moving any pebbles in the N_i configuration. We can assume that such a strategy is normal and regular by Corollary 2. We now consider Q_i, the i-th quantifier gadget.

1. Suppose $Q_i = \forall$. Suppose that t_0 is a time when s_i pebbles appear on the s_i-pyramid. After t_0, each of $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$ is only pebbled once before $\{q_i^1, \ldots, q_i^K\}$ are pebbled. Furthermore, by frugality, $\{a_i^1, \ldots, a_i^K\}$, $\{b_i^1, \ldots, b_i^K\}$, $\{c_i^1, \ldots, c_i^K\}$, $\{d_i^1, \ldots, d_i^K\}$, $\{f_i^1, \ldots, f_i^K\}$, and $\{g_i^1, \ldots, g_i^K\}$ are each pebbled only once after t_0 until $\{q_i^1, \ldots, q_i^K\}$ are pebbled. Let t_1 be the time when $x_i^{K'}$ is pebbled. Since our strategy is a normal and regular strategy, by Lemma 4 from t_1 until when $\{q_{i+1}^{1'}, \ldots, q_{i+1}^{K'}\}$ is pebbled, K pebbles remain on $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$, $\{x_i^1, \ldots, x_i^K\}$, or $\{f_i^1, \ldots, f_i^K\}$. From t_1 until $\{a_i^1, \ldots, a_i^K\}$ are pebbled, K pebbles are on $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$.

   To pebble a_i^1 requires pebbling $\{d_i^1, \ldots, d_i^K\}$. This requires removing all pebbles from the block except the K pebbles on $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$. By the normality of pebbling strategies, $\{d_i^1, \ldots, d_i^K\}$ are pebbled before everything else in the gadget besides $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$ and K pebbles remain on $\{d_i^1, \ldots, d_i^K\}$ until $\{b_i^1, \ldots, b_i^K\}$ are pebbled. To pebble $\{b_i^1, \ldots, b_i^K\}$ requires pebbling $\{c_i^1, \ldots, c_i^K\}$ which subsequently requires pebbling $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$. To pebble $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$ requires removing all pebbles on the gadget except those on $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$ and $\{d_i^1, \ldots, d_i^K\}$. Therefore, $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$ are pebbled immediately after $\{d_i^1, \ldots, d_i^K\}$ and K pebbles remain on $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$ or $\{\overline{x_i^1}, \ldots, \overline{x_i^K}\}$ until $\{c_i^1, \ldots, c_i^K\}$ are pebbled, which happens before $\{b_i^1, \ldots, b_i^K\}$ are pebbled. By normality, all pebbles except the ones on $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$ are removed from the connecting pyramids as soon as $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$ are pebbled. Let t_2 be the time the pebbles on the aforementioned pyramids are removed. Let t_3 be the first time after t_2 that $\{q_{i+1}^{1'}, \ldots, q_{i+1}^{K'}\}$ is pebbled.

   At t_2, there are pebbles on $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$, $\{d_i^1, \ldots, d_i^K\}$, and $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$. K pebbles must remain on $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$ and $\{d_i^1, \ldots, d_i^K\}$, each, until t_3. Furthermore, K pebbles must remain on either $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$ or $\{\overline{x_i^1}, \ldots, \overline{x_i^K}\}$ until t_3. First, suppose K pebbles remain on $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$ from t_2 to t_3. The configuration at t_2 is in N_{i+1} with a double false assignment to x_i, and none of the pebbles on the graph at t_2 can be removed until t_3. By the induction hypothesis, we pebble $\{q_{i+1}^{1'}, \ldots, q_{i+1}^{K'}\}$. Q_i can subsequently be pebbled with $3K$ pebbles that remained on the block up till t_3. Therefore, $\{q_i^1, \ldots, q_i^K\}$ can be pebbled with s_i additional pebbles with moving any of the pebbles in the N_i configuration. Since $Q_{i+1}x_{i+1} \cdots Q_n x_n F(e_1, \cdots, e_{2i-2}, False, False)$ is true then $\forall x_i Q_{i+1}x_{i+1} \cdots Q_n x_n F(e_1, \cdots, e_{2i-2})$ is also true.

In the case when the K pebbles on $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$ do not remain on $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$ until t_3, we argue that $\{q_{i+1}^{1'}, \ldots, q_{i+1}^{K'}\}$ must be pebbled twice, once with a false assignment to x_i and then with a true assignment to x_i. Either $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$ or $\{\overline{x_i^1}, \ldots, \overline{x_i^K}\}$ must be pebbled from t_2 to t_3. The only successors of $\{x_i^1, \ldots, x_i^K\}$ are $\{f_i^1, \ldots, f_i^K\}$ and $\{f_i^1, \ldots, f_i^K\}$ cannot be pebbled before t_3. Therefore, by regularity of pebbling strategies, we can arrange to move the K pebbles on $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$ to $\{\overline{x_i^1}, \ldots, \overline{x_i^K}\}$ in the time range $[t_2, t_2 + K^2]$ (and $t_3 > t_2 + K^2$) where they remain until t_3. The configuration at $t_2 + K^2$ is that N_{i+1} is assigned to the false assignment of x_i and none of the pebbles on the graph at $t_2 + K^2$ can be removed until t_3 by normality. By the induction hypothesis, $Q_{i+1}x_{i+1} \cdots Q_n x_n F(e_1, \cdots, e_{2i-2}, False, True)$ is true.

At t_3, there are pebbles on $\{d_i^1, \ldots, d_i^K\}$, $\{\overline{x_i^1}, \ldots, \overline{x_i^K}\}$, $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$ and $\{q_{i+1}^{1'}, \ldots, q_{i+1}^{K'}\}$. Vertices $\{q_i^1, \ldots, q_i^K\}$, $\{a_i^1, \ldots, a_i^K\}$, $\{b_i^1, \ldots, b_i^K\}$, $\{c_i^1, \ldots, c_i^K\}$, $\{f_i^1, \ldots, f_i^K\}$, and $\{g_i^1, \ldots, g_i^K\}$ are vacant because they cannot be pebbled before $\{q_{i+1}^{1'}, \ldots, q_{i+1}^{K'}\}$ are pebbled. Vertices $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$ couldn't have been repebbled between $t_2 + K^2$ and t_3 since $4K$ pebbles are fixed on $\{d_i^1, \ldots, d_i^K\}$, $\{\overline{x_i^1}, \ldots, \overline{x_i^K}\}$, $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$, and the paths from $\{p_0^1, \ldots, p_0^K\}$ to q_1 during that interval; thus $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$ and, by normality, the pyramids connected to the vertices cannot be pebbled in the interval. It does not matter whether there exists pebbles on $\{x_i^1, \ldots, x_i^K\}$ at t_3. We now show that immediately after t_3, a configuration in N_{i+1} with a true assignment to x_i is created, and that $\{q_{i+1}^{1'}, \ldots, q_{i+1}^{K'}\}$ must be repebbled while the pebbles in the configuration are fixed.

By frugality, the pebbles on $\{q_{i+1}^{1'}, \ldots, q_{i+1}^{K'}\}$ at t_3 remains until either $\{c_i^1, \ldots, c_i^K\}$ or $\{g_i^1, \ldots, g_i^K\}$ are pebbled. Vertices $\{q_{i+1}^{1'}, \ldots, q_{i+1}^{K'}\}$ cannot contain pebbles until $\{g_i^1, \ldots, g_i^K\}$ are pebbled since to pebble $\{g_i^1, \ldots, g_i^K\}$ requires all but $2K$ pebbles on the quantifier block or on the pyramids underneath $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$. K pebbles are fixed on $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$, $\{x_i^1, \ldots, x_i^K\}$, or $\{f_i^1, \ldots, f_i^K\}$ and K pebbles are fixed on $\{d_i^1, \ldots, d_i^K\}$, $\{b_i^1, \ldots, b_i^K\}$, or $\{a_i^1, \ldots, a_i^K\}$ until $\{q_i^1, \ldots, q_i^K\}$ is pebbled. Thus, the K pebbles on $\{q_{i+1}^{1'}, \ldots, q_{i+1}^{K'}\}$ at t_3 remain until $\{c_i^1, \ldots, c_i^K\}$ are pebbled and are removed before $\{g_i^1, \ldots, g_i^K\}$ are pebbled. Since $\{\overline{x_i^1}, \ldots, \overline{x_i^K}\}$ are pebbled at t_3 we can rearrange the strategy so that the K pebbles from pebbling $\{q_{i+1}^{1'}, \ldots, q_{i+1}^{K'}\}$ are moved to $\{c_i^1, \ldots, c_i^K\}$ by time $t_3 + K$.

Now the only successors of $\{c_i^1, \ldots, c_i^K\}$ and $\{b_i^1, \ldots, b_i^K\}$ are $\{b_i^1, \ldots, b_i^K\}$ and $\{a_i^1, \ldots, a_i^K\}$, respectively. Since $\{d_i^1, \ldots, d_i^K\}$ and $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$ both contain pebbles at $t_3 + K$, we can rearrange the strategy so that the pebbles on $\{c_i^1, \ldots, c_i^K\}$ are moved to $\{b_i^1, \ldots, b_i^K\}$ at $t_3 + 2K$ and to $\{a_i^1, \ldots, a_i^K\}$ at $t_3 + 3K$. K pebbles must remain on $\{a_i^1, \ldots, a_i^K\}$ until $\{q_i^1, \ldots, q_i^K\}$ are pebbled. Since $\{a_i^1, \ldots, a_i^K\}$ are only pebbled once after t_0 and before $\{q_i^1, \ldots, q_i^K\}$ are pebbled and are the only successors of $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$ aside from $\{x_i^1, \ldots, x_i^K\}$, we can further rearrange the strategy so that the pebbles on $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$ are moved to $\{x_i^1, \ldots, x_i^K\}$ at $t_3 + 3K + K^2$.

At $t_3 + 3K + K^2$, $\{a_i^1, \ldots, a_i^K\}$ contains K pebbles that will remain until $\{q_i^1, \ldots, q_i^K\}$ are pebbled, and $\{x_i^1, \ldots, x_i^K\}$ contain K pebbles that remain until $\{f_i^1, \ldots, f_i^K\}$ are pebbled. Vertices $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$ must be repebbled before $\{f_i^1, \ldots, f_i^K\}$ are pebbled, which must happen before $\{q_i^1, \ldots, q_i^K\}$ are pebbled. To pebble $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$, by Lemma 4, requires all the pebbles from this block except the ones on $\{a_i^1, \ldots, a_i^K\}$ and $\{x_i^1, \ldots, x_i^K\}$, so by normality $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$ are first pebbled after $t_3 + 3K + K^2$, and are each only pebbled once before

$\{f_i^1, \ldots, f_i^K\}$ are pebbled. Let t_4 be the time all the pebbles except the ones on $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$ are removed from the pyramids under the nodes where $t_4 > t_3 + 3K + K^2$. At t_4, there are pebbles on $\{a_i^1, \ldots, a_i^K\}$, $\{x_i^1, \ldots, x_i^K\}$, and $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$ and nowhere else on the i-th quantifier block. This configuration is in N_{i+1} with a true assignment to x_i and none of the pebbles on the graph at t_4 can be removed until after $\{q_{i+1}^{1'}, \ldots, q_{i+1}^{K'}\}$ are repebbled. By the induction hypothesis, $Q_{i+1}x_{i+1} \cdots Q_n x_n F(e_1, \cdots, e_{2i-2}, True, False)$ is true. Therefore, $\forall x_i Q_{i+1}x_{i+1} \cdots Q_n x_n F(e_1, \cdots, e_{2i-1})$ is true. This concludes the inductive step for the universal quantifier.

2. Suppose $Q_i = \exists$. Suppose t_1 is a time that $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$ is pebbled. By frugality, each of $\{a_i^1, \ldots, a_i^K\}$, $\{b_i^1, \ldots, b_i^K\}$, $\{c_i^1, \ldots, c_i^K\}$, $\{d_i^1, \ldots, d_i^K\}$, $\{e_i^1, \ldots, e_i^K\}$ and $\{q_{i+1}^{1'}, \ldots, q_{i+1}^{K'}\}$ are pebbled at most once after t_1 and before $\{q_i^1, \ldots, q_i^K\}$ are pebbled. Exactly as in the case of the universal quantifier, normality implies that $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$ are only pebbled once after t_1 and before $\{q_i^1, \ldots, q_i^K\}$ are pebbled, and are pebbled before anything else happens. K pebbles remain on $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$ or $\{x_i^1, \ldots, x_i^K\}$ until $\{a_i^1, \ldots, a_i^K\}$ are pebbled, and K pebbles remain on $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$ or $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$ until $\{b_i^1, \ldots, b_i^K\}$ are pebbled. To pebble $\{a_i^1, \ldots, a_i^K\}$ require pebbling $\{d_i^1, \ldots, d_i^K\}$, which require removing all pebbles from the block except the ones on $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$. Thus, $\{d_i^1, \ldots, d_i^K\}$ are pebbled before anything else except $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$, and pebbles remain on $\{d_i^1, \ldots, d_i^K\}$ until $\{c_i^1, \ldots, c_i^K\}$ are pebbled.

   To pebble $\{c_i^1, \ldots, c_i^K\}$ require pebbling $\{e_i^1, \ldots, e_i^K\}$ and hence $\{f_i^1, \ldots, f_i^K\}$. To pebble $\{f_i^1, \ldots, f_i^K\}$ requires removing all pebbles from this block except those on $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$ and $\{d_i^1, \ldots, d_i^K\}$. Thus, $\{f_i^1, \ldots, f_i^K\}$ are pebbled only once before $\{e_i^1, \ldots, e_i^K\}$ are pebbled, and this happens immediately after $\{d_i^1, \ldots, d_i^K\}$ are pebbled. K pebbles remain on $\{f_i^1, \ldots, f_i^K\}$, $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$, or $\{\overline{x_i^1}, \ldots, \overline{x_i^K}\}$ until $\{e_i^1, \ldots, e_i^K\}$ are pebbled. The only successors of $\{f_i^1, \ldots, f_i^K\}$ are $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$, and K pebbles remain on $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$ until $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$ are pebbled, so we can rearrange the strategy so that the first move after picking up the pebbles on the pyramids underneath $\{f_i^1, \ldots, f_i^K\}$ is to move the pebbles on $\{f_i^1, \ldots, f_i^K\}$ to $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$. Let t_2 be the time of this move, and let t_3 be the time $\{q_{i+1}^{1'}, \ldots, q_{i+1}^{K'}\}$ are pebbled. Note that since $\{f_i^1, \ldots, f_i^K\}$ are not repebbled between t_2 and t_3, neither are $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$. At t_2, there are pebbles on $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$, $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$, and $\{d_i^1, \ldots, d_i^K\}$ and until t_3, there must be pebbles on $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$ or $\{x_i^1, \ldots, x_i^K\}$, $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$ or $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$, $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$ or $\{\overline{x_i^1}, \ldots, \overline{x_i^K}\}$, and $\{d_i^1, \ldots, d_i^K\}$.

   Now we consider 3 cases. Suppose that the pebbles on $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$ are removed before t_3. Since the only successors of $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$ are $\{x_i^1, \ldots, x_i^K\}$ and $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$ and $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$ is not repebbled before t_3, we can rearrange the strategy so that the pebbles on $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$ are moved to $\{x_i^1, \ldots, x_i^K\}$ at $t_2 + K^2$. The configuration at $t_2 + K^2$ is then in N_{i+1} with the true assignment to x_i, and none of the pebbles can be removed until t_3. By the induction hypothesis, $Q_{i+1}x_{i+1} \cdots Q_n x_n F(e_1, \cdots, e_{2i-2}, True, False)$ is true.

   Suppose in the second case that K pebbles remain on $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$ until t_3, and the pebbles on $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$ are removed before t_3. We can rearrange the strategy so that the pebbles on $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$ are moved to $\{\overline{x_i^1}, \ldots, \overline{x_i^K}\}$ at $t_2 + K^2$. The configuration at $t_2 + K^2$ is in N_{i+1} with the false assignment to x_i, and no pebble can be removed until t_3. By the induction hypothesis, $Q_{i+1}x_{i+1} \cdots Q_n x_n F(e_1, \cdots, e_{2i-2}, False, True)$ is true.

Finally, suppose that pebbles remain on $\{x_i^{1\prime}, \ldots, x_i^{K\prime}\}$ and $\{\overline{x_i^{1\prime}}, \ldots, \overline{x_i^{K\prime}}\}$ until t_3. The configuration at t_2 is in N_{i+1} with a double false assignment to x_i, and no pebble is removed until t_3. By the induction hypothesis, $Q_{i+1}x_{i+1} \cdots Q_n x_n F(e_1, \cdots, e_{2i-2}, False, False)$ is true.

In each of the above cases, $\exists x_i Q_{i+1}x_{i+1} \cdots Q_n x_n F(e_1, \cdots, e_{2i-2})$ is true. This completes the inductive step for an existential quantifier, and the proof of the claim.

□

The proof of the above claim subsequently proves the lemma when $i = 1$. □

We now prove that $3Ku + 5K$ pebbles are necessary to pebble an unsatisfiable instance of QBF.

Lemma 7. *Given G which is constructed from the provided QBF instance, $B = Q_1x_1 \cdots Q_u x_u F$, using our modified reduction in Section 3.2.2 and Section 3.2.3, B is unsatisfiable if and only if $\mathsf{Peb}(G) \geq 3Ku + 5K$.*

Proof. We first prove that if B is unsatisfiable, then the number of pebbles necessary to pebble the modified construction requires at least $3Ku + 5K$ pebbles. By Lemma 4, there does not exist a frugal strategy such that $3Ku$ pebbles are not assigned to the u quantifier blocks when the clauses are pebbled and the number of pebbles used is less than $3Ku + 5K$. Therefore, there exists only $s - 3Ku \leq 4K + 1$ pebbles remaining to pebble the clauses assuming the player is given $3Ku + 4K + 1$ to begin with to pebble G. Provided $3Ku$ pebbles are on the quantifier blocks, Lemma 5 proves that at least $5K$ additional pebbles are needed to pebble one or more false clauses given that B is unsatisfiable.

Now we prove that if the number of pebbles necessary to pebble G is at least $3Ku + 5K$, then B is unsatisfiable. This proof is given by contradiction which immediately follows from Lemma 6. □

3.2.5 Proof of Inapproximability

Using Lemmas 6 and 7, we prove that it is PSPACE-hard to approximate the minimum number of black pebbles needed given a DAG, G, to an additive $n^{1/3-\varepsilon}$ for all $\varepsilon > 0$.

Lemma 8. *The number of nodes in G is $O(K^3(u^3 + c))$.*

Proof. By construction of G as defined in Sections 3.2.2 and 3.2.3, we create u variable gadgets, u quantifier blocks, and c clause gadgets. Each variable gadget contains $O(K^3)$ nodes since it contains two road graphs where each road graph with K width contains K^3 nodes. Each quantifier block contains a variable gadget and the pyramids that connect to the variable gadget and $O(K)$ other nodes. The total size of the pyramids is at most $\sum_{i=1}^{3Ku+4K+1} i^2 = O(K^3u^3)$. Therefore, the total size of all the quantifier blocks is $O(K^3u^3)$ since the total size of the quantifier blocks without the connecting pyramids is $O(K^3u)$ and the total size of all the pyramids is $O(K^3u^3)$.

The clauses each have size $O(K^2)$ since the clauses solely consist of a constant number of pyramids of height $O(K)$. Therefore, the total size of all the clauses is $O(K^2c)$.

Thus, G has $O(K^3(u^3 + c))$ number of nodes in total. □

Theorem 4 (Restatement of Theorem 1)**.** *The minimum number of pebbles needed in the standard pebble game on DAGs with maximum indegree 2 is PSPACE-hard to approximate to additive $n^{1/3-\varepsilon}$ for $\varepsilon > 0$.*

Proof. From Lemmas 6 and 7, the cost of pebbling a graph constructed from a satisfiable B is at most $3Ku + 4K + 1$ whereas the cost of pebbling a graph constructed from an unsatisfiable B is at least $3Ku + 5K$.

As we can see, the aforementioned reduction is a gap-producing reduction with a gap of $K - 1$ pebbles. Then, all that remains to be shown is that for any $\varepsilon > 0$, it is the case that $K \geq (K^3(u^3 + c))^{(1/3-\varepsilon)}$. (Note that for $\varepsilon > 1/3$, setting K to any positive integer achieves this bound.) Suppose we set $K = \max(u, c)^a$ where $a > 0$. We show that $K = (K^3[\max(u, c)]^3)^{(1/3-\varepsilon)}$ for some valid setting of a for every $0 < \varepsilon \leq 1/3$. Solving for a in terms of ε gives us $a = \frac{1}{3\varepsilon} - 1 \geq 0$ when $\varepsilon \leq 1/3$ and is finite when $\varepsilon > 0$.

For values of $a \geq 0$, we can duplicate the clauses and variables gadgets so that u and c are large enough such that $K = \max(u, c)^a \geq 2$. Let $d = \max(u, c)$. Then, we need d to be large enough so that $d^a \geq 2$ (i.e. we want d^a to be some integer). Then, we can set $d \geq 2^{1/a}$. Thus, we can duplicate the number of variables and clauses so that $d \geq 2^{\frac{3\varepsilon}{1-3\varepsilon}}$. (Note that for cases when a is very small, e.g. $a = 0$ when $\varepsilon = 1/3$, any constant K would suffice.)

Therefore, for every $\varepsilon > 0$, we can construct a graph with a specific K calculated from ε such that it is PSPACE-hard to find an approximation within an additive $n^{1/3-\varepsilon}$ where n is the number of nodes in the graph. □

4 Hard to Pebble Graphs for Constant k Pebbles

It is long known that the maximum number of moves necessary to pebble any graph with constant k pebbles is $O(n^k)$. (Note that the maximum number of moves necessary to pebble any graph is either $O(n^{k-1})$ or $O(n^k)$ depending on whether or not sliding is allowed. Here, we allow sliding in all of our games. The bound of $O(n^{k-1})$ proven in [Nor15] is one for the case when sliding is not allowed.) The upper bound of $O(n^k)$ for any constant k number of pebbles submits to a simple combinatorial proof adapted from [Nor15] to account for sliding. However, to the best of the author's knowledge, examples of such families of graphs that require $O(n^k)$ moves to pebble using k pebbles did not exist until very recently in an independent work [AdRNV17]. In this section, we present an independent, simple to construct family of graphs that require $\Theta(n^k)$ time for constant k number of pebbles in both the standard and black-white pebble games. We further reduce the indegree of nodes in this family of graphs to 2 and show that our results still hold. Furthermore, we show this family of graphs to exhibit a steep time-space trade-off (from exponential in k to linear) even when k is not constant. Such families of graphs could potentially have useful applications in cryptography in the domain of proofs of space and memory-hard functions [AS15].

We construct the following family of graphs, $\mathbb{H}$, below and show that for constant k pebbles, the number of steps it takes to pebble the graph $H_{n,k} \in \mathbb{H}$ with k pebbles and n nodes is $\Omega(n^k)$. We also show a family of graphs, $\mathbb{H}_2$ with indegree 2 that exhibits the same asymptotic tradeoff.

We construct the family of graphs $\mathbb{H}$ with arbitrary indegree in the following way.

Definition 4. *Given a set of n nodes and maximum number of pebbles k where $k < \sqrt{n}$, we lexicographically order the nodes (from 1 to n) and create the following set of edges between the nodes where directed edges are directed from v_i to v_j where $i < j$. Let $[n]$ be the ordered set $[1, \ldots, n]$:*

1. *v_i and v_{i+1} for all $i \in [k-1, n]$*
2. *v_i and v_j for all $i \in [l-1]$ for all $2 \leq l \leq k$ and $j \in \{f(l) + 2r - 2\}$ for all $r \in [\frac{n-k}{2k}]$ where $f(l) = (k-1) + (l-1)(\frac{n-k}{k}) + 2$.*

3. v_i and v_j for all $i = f(l)-2$ and $j \in \{f(l)+2r-1\}$ for all $r \in [0, \frac{n-k}{2k}-1]$ where $l \in [1, k-1]$.

The target node (the only sink) is v_n. Note that the sources in our construction are v_j for all $j \in [1, k-1]$.

Below is an example graph (Fig. 12) in our family when $k = 5$ and $n = 54$.

We now prove the time bound for this family of graphs $\mathbb{H}$ for all $k < \sqrt{n}$.

To prove the minimum number of pebbles necessary to pebble the graph, it is sufficient to study the number of blocked paths in any graph G [Nor15]. We define a *blocked path* as in [Nor15].

Definition 5 (Blocked Paths [Nor15])**.** *A set of vertices, U, blocks a path, P, if $U \cap P \neq \emptyset$. U blocks a set of paths $\mathbb{P}$ if U blocks P for all $P \in \mathbb{P}$.*

Lemma 9. *The minimum number of pebbles necessary to pebble $H_{n,k} \in \mathbb{H}$ is k.*

Proof. The degree of the graph $H_{n,k}$ is k, therefore, at least k pebbles are necessary to pebble $H_{n,k} \in \mathbb{H}$. □

Theorem 5. *The number of moves necessary to pebble $H_{n,k}$ is $\Theta((\frac{n-k}{2k})^k)$ for $k < \sqrt{n}$.*

Proof. Let $\mathcal{P} = \{P_0, \ldots, P_a\}$ where $a = \Theta((\frac{n-k}{2k})^k)$. Suppose that at time t, there exists at least k paths from sources to targets which are blocked by k pebbles placed on the graph. By Lemma 9 and by construction of $H_{n,k}$, we know t must exist at some point in the pebbling of $H_{n,k}$. Let v_i be a degree-k node to be pebbled. Let t be the time step immediately before v_i is pebbled. Then, $|P_t| = k$ and there exists k blocked paths from each $u \in \mathsf{pred}(v_i)$ that is a source and v_k to the target v_n. Each of the paths is blocked by a pebble on $\mathsf{pred}(v_i)$ at time t.

Let $t_{now} = t + 1$ be the time when v_i is pebbled. After v_i has been pebbled, there are two different paths going from the sources through v_{i+1}. However, none of the pebbles placed at time $t' \in [t, t_{now}]$ blocks the path from v_k to v_n that does not include v_i. Therefore, a pebble must be placed on this path to block it, resulting in repebbling $v_{(k-1)+(k-1)(\frac{n-k}{k})}$.

By induction on the level number, $1 \leq l \leq k$, where $T(1) = \frac{n-k}{k}$ is the base case when node $v_{(k-1)+\frac{n-k}{k}}$ is pebbled. We see by the above argument that the resulting number of moves is $T(l) = (\frac{n-k'}{2k'})T(l-1) + l(\frac{n-k'}{2k'})$ where $k' = k$ is not a variable in the recursion (i.e. not changing) and we compute $T(k)$. Therefore,

$$T(k) = (\frac{n-k}{2k})^k + \sum_{i=0}^{k-1}(k-i)(\frac{n-k}{2k})^{i+1} \geq 2(\frac{n-k}{2k})^k = \Theta((\frac{n-k}{2k})^k).$$

□

We obtain the following when k is constant:

Corollary 3 ($H_{n,k}$: $\Omega(n^k)$ moves bound for constant k)**.** *When k is constant, pebbling $H_{n,k} \in \mathbb{H}$ using k pebbles takes $\Omega(n^k)$ time.*

This result of itself partially answers a longstanding open question posed in [Nor15] whether a family of graphs with constant degree can have a number of moves, $\Omega(n^k)$, that meets the upper bound for constant k number of pebbles. Now, we completely resolve this open question by proving

that the bound also holds for the black-white pebble game using our independent construction from [AdRNV17].

Before we prove the result, we quickly give the rules for the black-white pebble game as promised in Section 1.

Black-White Pebble Game Rules:

BLACK-WHITE PEBBLE GAME

Input: Given a DAG, $G = (V, E)$. Let $\mathsf{pred}(v) = \{u \in V : (u, v) \in E\}$. Let $S \subseteq V$ be the set of sources of G and $T \subseteq V$ be the set of targets of G. Let $\mathcal{P} = \{P_0, \ldots, P_\tau\}$, where $P_i = (B_i, W_i)$ (B_t is the set of nodes with black pebbles and W_t is the set of nodes with white pebbles), be a valid pebbling strategy that obeys the following rules where $P_0 = (\emptyset, \emptyset)$ and $P_\tau = (T, \emptyset)$. Let $\mathsf{Peb}(G, \mathcal{P}) = \max_{i \in [\tau]}\{|P_i|\}$ where $|P_i| = |B_i| + |W_i|$.

Rules:

1. At most one pebble can be placed or removed from a node at a time.
2. A black pebble can be placed on any source, $s \in S$. A white pebble can always be removed from a source.
3. A black pebble can be removed from any vertex. A white pebble can be removed from a non-source vertex, v, at time i if and only if $\mathsf{pred}(v) \in P_{i-1}$.
4. A black pebble can be placed on a non-source vertex, v, at time i if and only if $\mathsf{pred}(v) \in P_{i-1}$. A white pebble can be placed on an empty vertex at any time.
5. A black pebble can be moved from vertex v to vertex w at time i if and only if $(v, w) \in E$ and $\mathsf{pred}(w) \in P_{i-1}$. A white pebble can be moved from vertex w to vertex v if and only if $(v, w) \in E$ and $\mathsf{pred}(w) \backslash v \in P_{i-1}$ and $v \notin P_{i-1}$.

Goal: Determine $\min_{\mathcal{P}}\{\mathsf{Peb}(G, \mathcal{P})\}$ using a valid strategy $\mathcal{P}$.

Theorem 6 ($H_{n,k}$: $\Omega(n^k)$ moves bound black-white pebble game)**.** *The number of moves necessary to pebble $H_{n,k}$ is $\Theta((\frac{n-k}{2k})^k)$ for $k < \sqrt{n}$ using the rules of the black-white pebble game.*

Proof. Let $\mathcal{P} = \{P_0, \ldots, P_a\}$ be a black-white pebbling strategy where $a = \Theta((\frac{n-k}{2k})^k)$. At least k pebbles must be used to pebble nodes with indegree k. We proved in Theorem 5 that strategies that use only black pebbles must make $\Theta((\frac{n-k}{2k})^k)$ moves. Therefore, in order to use fewer than $\Theta((\frac{n-k}{2k})^k)$ moves, at least one white pebble must be used.

Let $T_{bw}(n, k)$ be the minimum time of pebbling $H_{n,k}$ using k black and white pebbles. $T_{bw}(\frac{n-k}{k} + 1, 1) \geq \frac{n-k}{k} + 1$. Because of the recursive structure of the graph family, we now show that

$$T_{bw}(n, l) \geq (\frac{n-k}{2k})T_{bw}((l-1)(\frac{n-k}{k}) + l - 1, l - 1) + l(\frac{n-k}{2k})$$

even when using black and white pebbles. Solving for $T(k)$ gives the number of moves to pebble $H_{n,k}$ using the rules of the black-white pebble game. The proof of the theorem then follows directly from the base case and the proof of Theorem 5.

To show the above, we first consider the case when v_n is pebbled with a white pebble. In this case, the predecessors of v_n must be pebbled with either black or white pebbles. v_n must be repebbled with a black pebble after the predecessors are pebbled, resulting in a strategy that does not use the minimum number of moves. Therefore, v_n is never pebbled with a white pebble. Now consider $\mathsf{pred}(v_n)$. If v_n has k predecessors, suppose the non-source predecessor of v_n is pebbled at t_1. Then, the earliest that v_n can be pebbled is at time $t_1 + k - 1$ regardless whether or not black or white pebbles are used. Given that k pebbles must be used to pebble v_{n-i} where i is even and v_{n-i} has k predecessors, $v_{k+(k-1)(\frac{n-k}{k})}$ must be pebbled $\frac{n-k}{2k}$ times. Thus, we have shown that regardless of whether black or white pebbles are used, we reduce to the above recursive relation. □

4.1 Max Indegree-2 Hard to Pebble Graphs

If we modify the construction presented in Definition 4 such that every node of degree $d > 2$ is replaced with a pyramid of height d, then we obtain the results we would like for the standard pebble game taking $\Omega(n^k)$ moves for any n and constant k. Rather than creating a unique pyramid for each node of degree $d > 2$, we create one height k pyramid Π_h and connect it to our construction, described below. From this construction, we obtain Theorem 2 as stated in the introduction.

Definition 6 (Standard Pebbling Construction with Max Indegree-2)**.** *We create the max indegree-2 hard to pebble family of graphs using the standard pebble game as follows. Suppose we have a total of n nodes.*

1. *Create a height $k-1$ pyramid and label the roots of pyramids of heights in the range $i \in [1, k-1]$, r_i.*
2. *Sort the remaining $n - \frac{(k-1)k}{2}$ vertices and create edges (v_i, v_j) where $i < j$ in the sorted order.*
3. *Create edges (r_i, v_j) for all $i \in [1, l-1]$ for all $1 \leq l \leq k$ and $j = f(l) + i + g + 1$ for all $g \in [\frac{n-\frac{(k-1)k}{2}}{kl}]$ and $f(l) = \frac{(k-1)(k)}{2} + (l-1)(\frac{n-\frac{(k-1)k}{2}}{k})$.*
4. *Create edges (v_i, v_j) for all $i = f(l) - 1$ and $j \in \{f(l) + gl\}$ for all $g \in [0, \frac{n-\frac{(k-1)k}{2}}{kl}]$ where $l \in [1, k-1]$.*

The target node is v_n.

Theorem 7. *There exists a family of graphs with n vertices and maximum indegree 2 such that $\Omega((\frac{n-k^2}{k^2})^k)$ moves are necessary to pebble any graph with n vertices in the family using $k < \sqrt{\frac{n}{2}}$ pebbles in the standard pebble game.*

Proof. By Definition 6, creating the pyramid of height $k-1$ requires $O(k^2)$ nodes. It suffices to only create one pyramid since a pyramid of any height less than $k-1$ is contained in a pyramid of height $k-1$. Furthermore, considering that the same k nodes are used for all nodes in different columns in Theorems 5 and 6, it does not matter that the different pyramids share nodes. For every node of degree $d \geq 3$ in the construction defined by Definition 4, we replace the node by a path with nodes connected to pyramids of heights, $h \in [1, d-1]$. By normality, for any pyramid of height h, we must remove h pebbles from the graph to pebble it in the standard pebble game. For paths that connect to pyramids of height $h \in [1, k-1]$, there are only two ways to pebble the pyramid of

height i. Either a pebble remains on the apex of the pyramid of height i or i pebbles are removed from the graph to pebble the pyramid using a normal strategy. To pebble the path connected to pyramids of heights $h \in [1, k-1]$ requires a total of $k-1$ pebbles either remaining on the pyramids or removed from the graph to be used to pebble the apex of each of the pyramids.

Suppose $T(l)$ is the time of pebbling the last node in the topological order of layer l with base case $T(1) = \frac{n-\frac{(k-1)k}{2}}{k}$. Then, we obtain the following recursive equation from our construction in Definition 6:

$$T(l) = \frac{n-\frac{(k-1)k}{2}}{kl}T(l-1) + (l-1)\Theta(\frac{n-k^2}{kl}) = \Omega((\frac{n-k^2}{k^2})^k)$$

□

The proof of number of standard pebbling moves necessary in pebbling the family of graphs defined by Definition 6 is $\Omega(n^k)$ when k is constant, proving part of Theorem 2.

To prove the result for the black-white pebble game, we use a result from [Lou79] and [Nor15] that gives a precise space cost for a complete binary search tree.

Theorem 8 (Black-White Pyramid Pebble Price [Lou79, Nor15])**.** *For a complete binary tree T_h of height $h \geq 1$ it holds that the black-white persistent pebbling cost is $\lfloor \frac{h+3}{2} \rfloor$. The* persistent *pebbling cost is defined as the cost of pebbling the root of T_h with a black pebble that remains on the root.*

We state as an immediate corollary of Theorem 8:

Corollary 4. *For a complete binary tree T_h of height $h \geq 1$ where $h \mod 2 = 1$, the cost of pebbling the root of T_h in the first step using a white pebble is $\lfloor \frac{h+3}{2} \rfloor$.*

Proof. Suppose for the sake of contradiction that the cost of pebbling the root of T_h in the first step using a white pebble is less than $\lfloor \frac{h+3}{2} \rfloor$, then according to the algorithm presented in [Lou79], the cost of persistently pebbling a binary tree of height h' where $h' = h+1$ is equal to $\lfloor \frac{h'+3}{2} \rfloor$ since the original strategy of pebbling one predecessor with a black pebble persistently and the other one with a white pebble results in the persistent cost to be $\lfloor \frac{h'+3}{2} \rfloor$. However, this contradicts with the stated lower bound of $\lceil \frac{h'+3}{2} \rceil = \lfloor \frac{h'+3}{2} \rfloor + 1$ [Lou79] since $h' \mod 2 = 0$. □

Using Theorem 8 and Corollary 4, we can define a class of graphs very similar to the class of graphs defined by Definition 6.

Definition 7 (Black-White Pebbling Construction with Max Indegree-2)**.** *We create the max indegree-2 hard to pebble family of graphs using the black-white pebble game as follows. Suppose we have a total of n nodes.*

1. *Create a height $H = 2(k-1) - 3 = 2k - 5$ complete binary tree and label the roots of trees of heights $2i - 3$ for $i \in [1, k-1]$, r_i.*
2. *Sort the remaining $n - 2^{2k-5}$ vertices and create edges (v_i, v_j) where $i < j$ in the sorted order.*
3. *Create edges (r_i, v_j) for all $i \in [1, l-1]$ for all $1 \leq l \leq k$ and $j = f(l) + i + g + 1$ for all $g \in [\frac{n-2^{2k-5}}{kl}]$ and $f(l) = 2^{2k-5} + (l-1)(\frac{n-2^{2k-5}}{k})$.*

4. *Create edges* (v_i, v_j) *for all* $i = f(l) - 1$ *and* $j \in \{f(l) + gl\}$ *for all* $g \in [0, \frac{n-2^{2k-5}}{kl}]$ *where* $l \in [1, k-1]$.

The target node is v_n.

Now we can prove our main theorem for black-white pebbling of our modified graph class as defined in Definition 7.

Theorem 9. *There exists a family of graphs with* n *vertices and maximum indegree* 2 *such that* $\Omega((\frac{n-2^{(2k-5)}}{k^2})^k)$ *moves are necessary to pebble any graph with* n *vertices in the family using* $k = o(\log n)$ *pebbles in the black-white pebble game.*

Proof. We create one complete binary tree of height $2k-5$. For every node of degree $d > 2$, we create a path where each node in the path is connected to roots of trees of heights $2i-3$ for all $i \in [1, d-1]$. As in the proof for Theorem 7, $d-1$ pebbles in total must either be on the roots of the trees or removed from the graph to pebble the roots of these trees regardless of whether black or white pebbles are used (by Theorem 8 and Corollary 4).

This will ensure that all d pebbles are used to pebble the binary search trees and the last node of the path. Therefore, we reach a similar recursive equation as in Theorem 7, using $T(l)$ as the time cost of pebbling level l with base case $T(1) = \frac{n-2^{2k-5}}{k}$:

$$T(l) = \frac{n-2^{2k-5}}{kl} T(l-1) + (l-1)\Theta(\frac{n-2^{2k-5}}{kl}) = \Omega((\frac{n-2^{2k-5}}{k^2})^k)$$

□

The proof of number of black-white pebbling moves necessary in pebbling the family of graphs defined by Definition 7 is $\Omega(n^k)$ when k is constant, concluding the proof of Theorem 2.

5 Open Problems

There are a number of open questions that naturally follow the content of this paper.

The first obvious open question is whether the techniques introduced in this paper can be tweaked to allow for a PSPACE-hardness of approximation to an $n^{1-\varepsilon}$ additive factor for any $\varepsilon > 0$. We note that the trivial method of attempting to reduce the size of the subgraph gadgets used in the variables (i.e. use a different construction than the road graph such that less than K^3 nodes are used) is not sufficient since the number of nodes in the graph is still $\Theta(K^3(u^3 + c))$. This is not to say that such an approach is not possible; simply that more changes need to be made to all of the other gadgets. The next logical step is to determine whether $\mathsf{Peb}(G)$ can be approximated to a constant 2 factor multiplicative approximation.

Another open question is whether the techniques introduced in this paper can be applied to show hardness of approximation results for other pebble games such as the black-white or reversible pebble games. The main open question in the topic of hardness of approximation of pebble games is whether these pebble games can be approximated to any multiplicative factors smaller than $n/\log n$ or whether the games are PSPACE-hard to approximate to any constant factor, perhaps even logarithmic factors.

With regard to hard to pebble graphs, we wonder if our graph family could be improved to show $\Omega(n^k)$ for any $0 < k \leq n/\log n$. This would be interesting because to the best of the authors' knowledge we do not yet know of any graph families that exhibit sharp (asymptotically tight) time-space trade-offs for this entire range of pebble number.

We also reiterate the persistent black-white pebbling cost of a pyramid (an open problem presented in [Nor15]) is an interesting open problem with respective to our results because it would broaden the range of allowed k in Theorem 9.

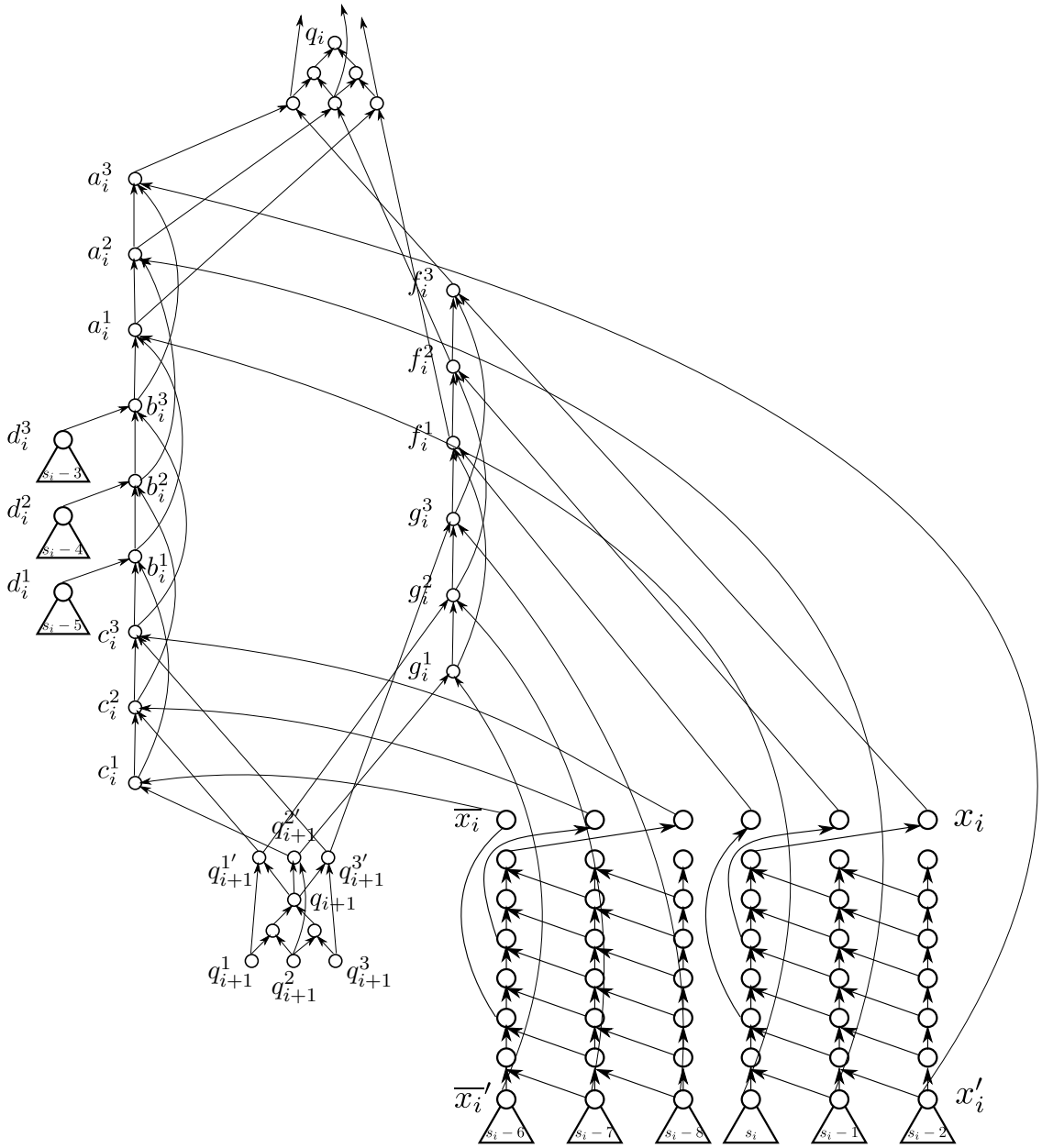

Figure 8: Modified universal quantifier block. Here $K = 3$.

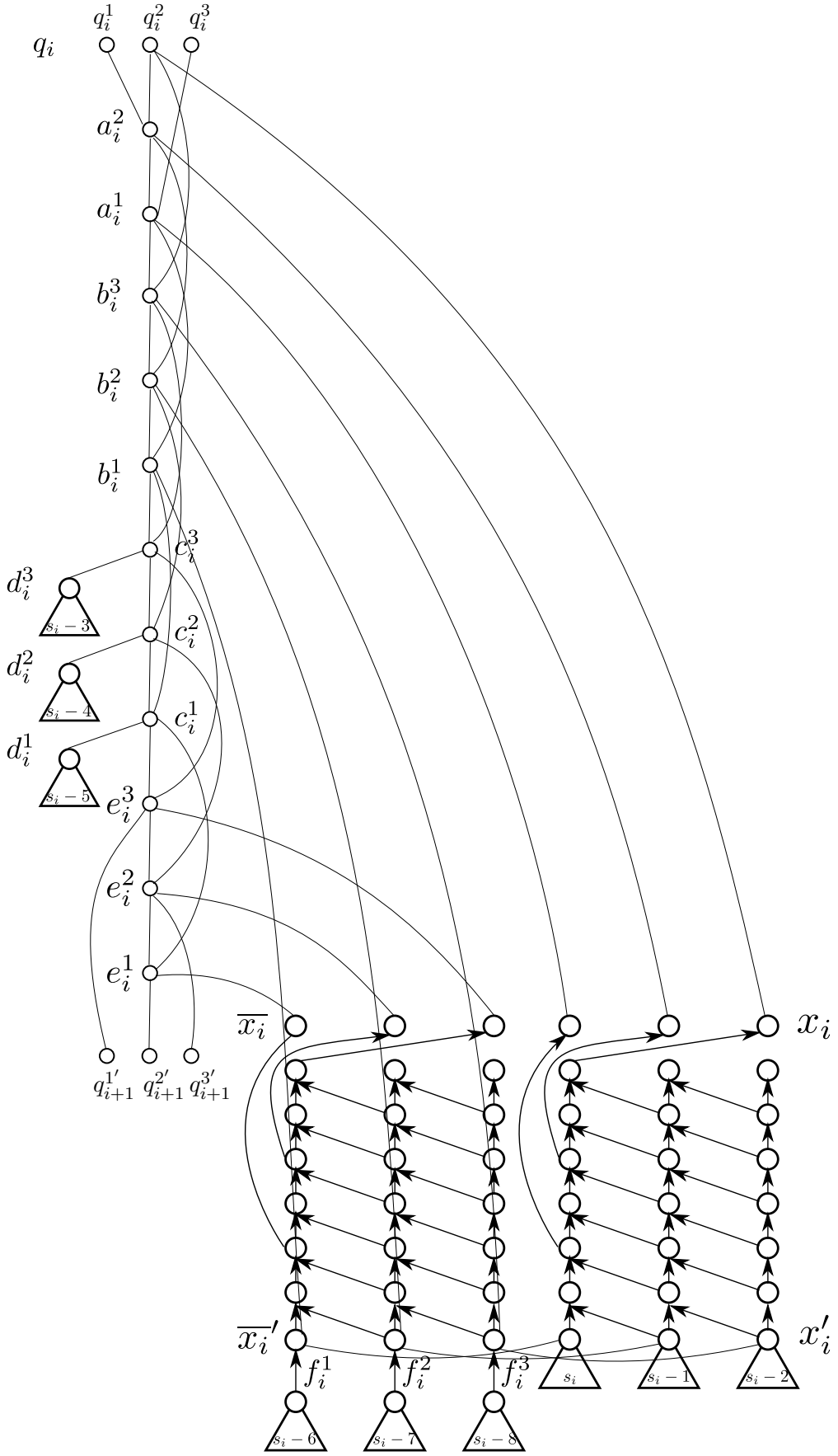

Figure 9: Modified existential quantifier block. Here $K = 3$.

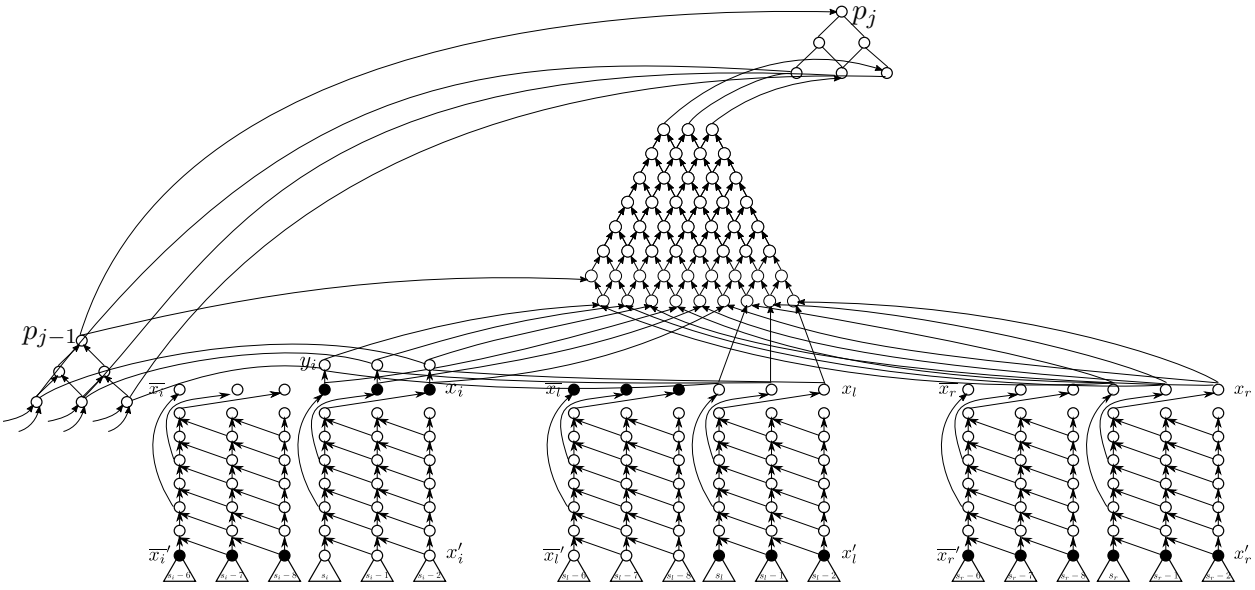

Figure 10: Modified clause gadget. The clause here is (x_i, x_l, x_r) where $x_i = True$, $x_l = False$, and $x_r = False$. Here $K = 3$.

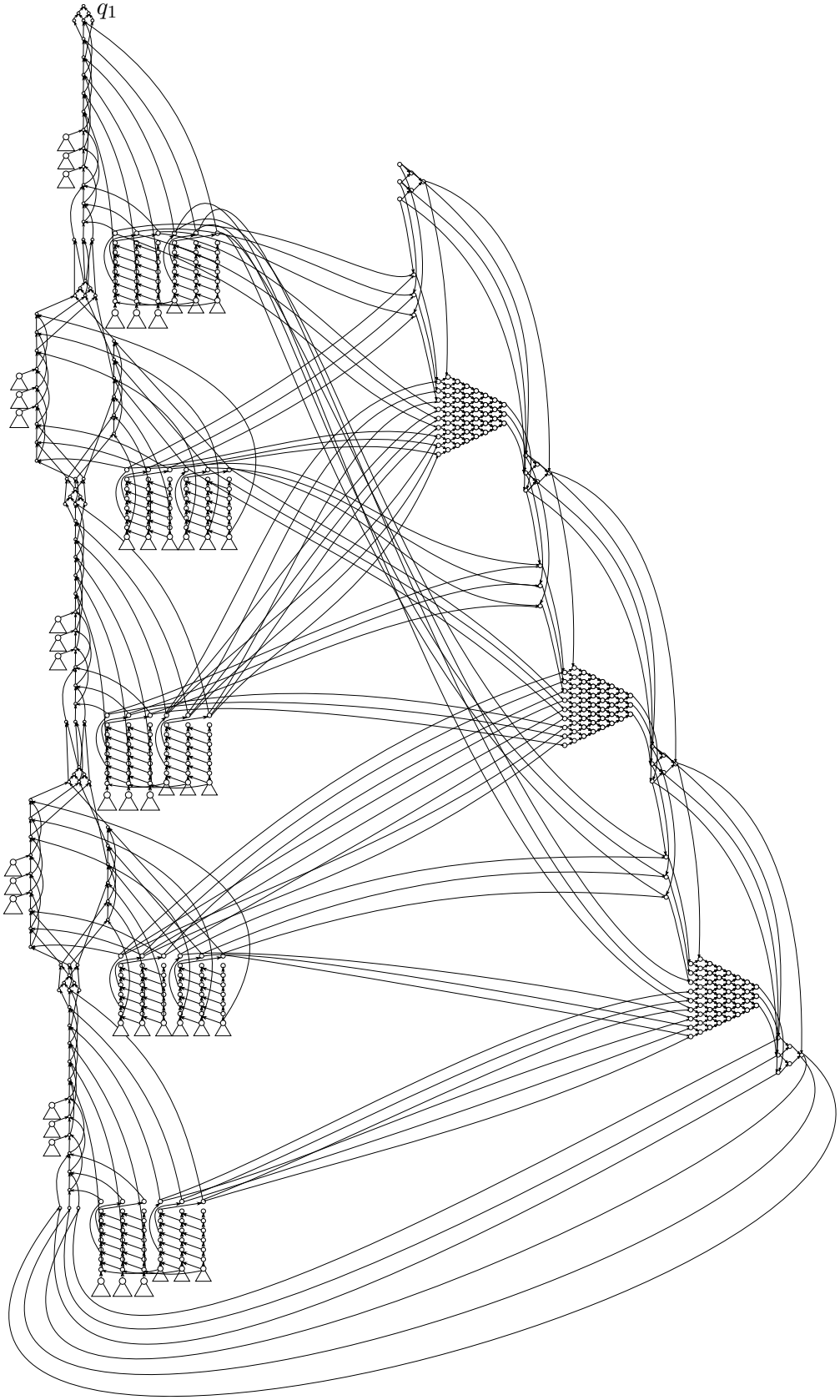

Figure 11: Modified full construction. Duplicate first clause has been left out for clarity.

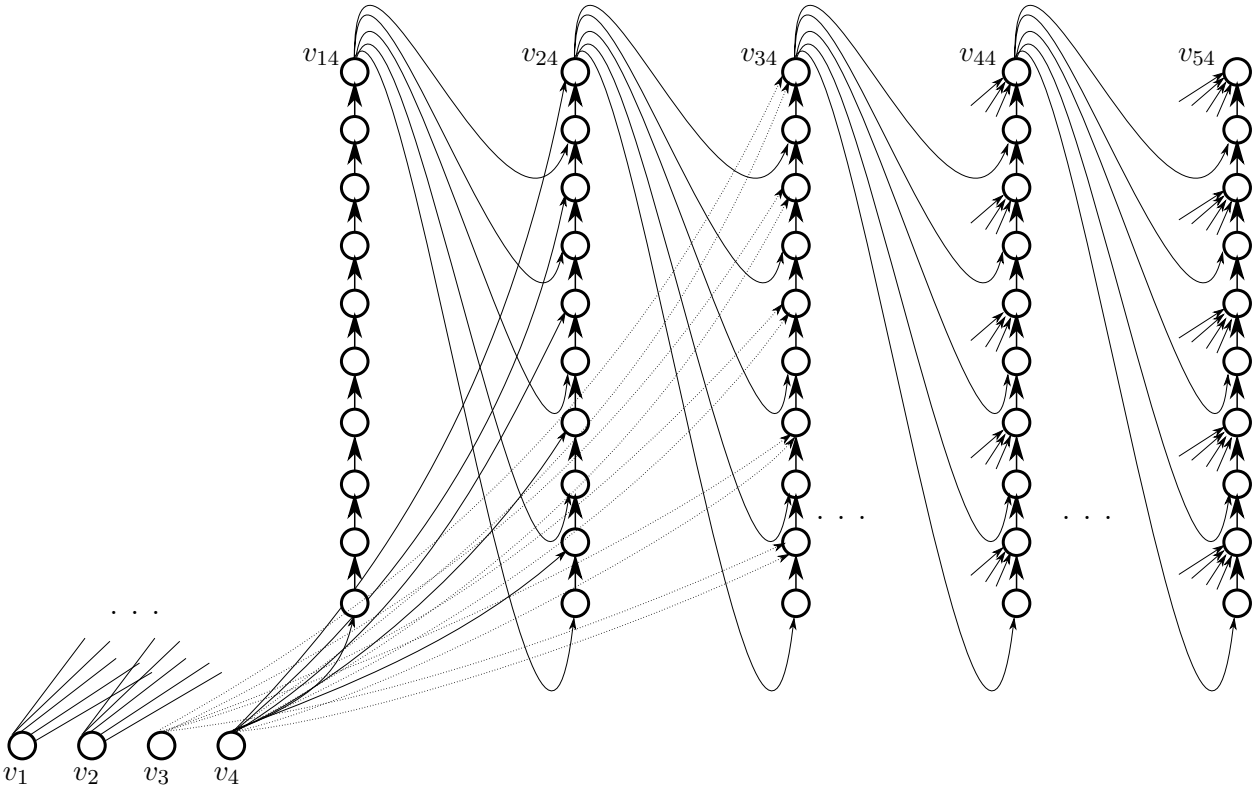

Figure 12: Example member of class of graphs where $k = 5$ and $n = 54$. One can make any graph of this class into an indegree-2 graph by replacing the input vertices by pyramids and performing the modifications described in Section 4.1.

References

[AdRNV17] Joël Alwen, Susanna F. de Rezende, Jakob Nordström, and Marc Vinyals. Cumulative space in black-white pebbling and resolution. In *Innovations in Theoretical Computer Science, ITCS 2017, Berkeley, CA, USA, 9-11 January, 2017*, 2017.

[AS15] Joël Alwen and Vladimir Serbinenko. High parallel complexity graphs and memory-hard functions. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 595–603, 2015.

[Ben89] Charles H. Bennett. Time/space trade-offs for reversible computation. *SIAM J. Comput.*, 18(4):766–776, August 1989.

[CLNV15] Siu Man Chan, Massimo Lauria, Jakob Nordström, and Marc Vinyals. Hardness of approximation in PSPACE and separation results for pebble games. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 466–485, 2015.

[CS74] Stephen Cook and Ravi Sethi. Storage requirements for deterministic / polynomial time recognizable languages. In *Proceedings of the Sixth Annual ACM Symposium on Theory of Computing*, STOC '74, pages 33–39, New York, NY, USA, 1974. ACM.

[EBL79] Peter Emde Boas and Jan Leeuwen. *Theoretical Computer Science 4th GI Conference: Aachen, March 26–28, 1979*, chapter Move rules and trade-offs in the pebble game, pages 101–112. Springer Berlin Heidelberg, Berlin, Heidelberg, 1979.

[GLT79] John R. Gilbert, Thomas Lengauer, and Robert Endre Tarjan. The pebbling problem is complete in polynomial space. In *Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing*, STOC '79, pages 237–248, New York, NY, USA, 1979. ACM.

[GT78] John R. Gilbert and Robert E Tarjan. Variations of a pebble game on graphs. Technical report, Stanford, CA, USA, 1978.

[HP10] Philipp Hertel and Toniann Pitassi. The PSPACE-completeness of black-white pebbling. *SIAM J. Comput.*, 39(6):2622–2682, April 2010.

[HPV77] John Hopcroft, Wolfgang Paul, and Leslie Valiant. On time versus space. *J. ACM*, 24(2):332–337, April 1977.

[JWK81] Hong Jia-Wei and H. T. Kung. I/O complexity: The red-blue pebble game. In *Proceedings of the Thirteenth Annual ACM Symposium on Theory of Computing*, STOC '81, pages 326–333, New York, NY, USA, 1981. ACM.

[Lou79] M.C. Loui. *The Space Complexity of Two Pebble Games on Trees*. Technical memoranda. Mass. Inst. of Technology, Laboratory for Computer Science, 1979.

[LT79] Thomas Lengauer and Robert Endre Tarjan. Upper and lower bounds on time-space tradeoffs. In *Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing*, STOC '79, pages 262–277, New York, NY, USA, 1979. ACM.

[Nor15] Jakob Nordstrom. New wine into old wineskins: A survey of some pebbling classics with supplemental results. 2015.

[PTC76] Wolfgang J. Paul, Robert Endre Tarjan, and James R. Celoni. Space bounds for a game on graphs. In *Proceedings of the Eighth Annual ACM Symposium on Theory of Computing*, STOC '76, pages 149–160, New York, NY, USA, 1976. ACM.

[Set75] Ravi Sethi. Complete register allocation problems. *SIAM J. Comput.*, 4(3):226–248, 1975.