Cumulative Space in Black-White Pebbling and Resolution*

Joël Alwen[1], Susanna F. de Rezende[2], Jakob Nordström[3], and Marc Vinyals[4]

1 **IST Austria, Vienna, Austria**
`jalwen@ist.ac.at`
2 **KTH Royal Institute of Technology, Stockholm, Sweden**
`sfdr@kth.se`
3 **KTH Royal Institute of Technology, Stockholm, Sweden**
`jakobn@kth.se`
4 **KTH Royal Institute of Technology, Stockholm, Sweden**
`vinyals@kth.se`

Abstract

We study space complexity and time-space trade-offs with a focus not on peak memory usage but on overall memory consumption throughout the computation. Such a cumulative space measure was introduced for the computational model of parallel black pebbling by [Alwen and Serbinenko 2015] as a tool for obtaining results in cryptography. We consider instead the non-deterministic black-white pebble game and prove optimal cumulative space lower bounds and trade-offs, where in order to minimize pebbling time the space has to remain large during a significant fraction of the pebbling.

We also initiate the study of cumulative space in proof complexity, an area where other space complexity measures have been extensively studied during the last 10–15 years. Using and extending the connection between proof complexity and pebble games in [Ben-Sasson and Nordström 2008, 2011], we obtain several strong cumulative space results for (even parallel versions of) the resolution proof system, and outline some possible future directions of study of this, in our opinion, natural and interesting space measure.

1998 ACM Subject Classification F.1.3 Complexity Measures and Classes – Relations among complexity measuresm, F.4.1 Mathematical Logic – Computational logic, F.2.2 Nonnumerical Algorithms and Problems – Complexity of proof procedures

Keywords and phrases pebble game, pebbling, proof complexity, space, cumulative space, clause space, resolution, parallel resolution

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.38

1 Introduction

The time and space complexity measures are at the heart of understanding computation. Unfortunately, there is little we can say about general computation models such as Boolean circuits, let alone Turing machines. But if we allow ourselves to work with simpler models of computation, then we have a better chance at understanding these resources, and in fact there has been impressive progress in restricted models like bounded-depth circuits.

* The second, third and fourth authors were funded by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007–2013) / ERC grant agreement no. 279611. The second author was also supported by Swedish Research Council grants 621-2010-4797 and 621-2012-5645.

One of the first success stories in this direction are pebble games. The original *(black) pebble game* is played by a single player on a directed acyclic graph (DAG) with a single sink and all vertices having bounded indegree and consists of two simple rules:

1. we can add a pebble to a vertex if all its direct predecessors have pebbles, and
2. we can remove a pebble from a vertex at any time.

The goal of the game is to place a pebble on the sink of the graph. Time is measured as the number of moves to reach this goal, and the space is the maximum number of pebbles needed simultaneously at any point during the pebbling.

Quite surprisingly, this seemingly simple and innocent game can be used to obtain strong results even for general computation models, as it is at the core of the $\mathsf{DTIME}(t) \subseteq \mathsf{SPACE}(t/\log t)$ space upper bound for Turing machines in [33]. Pebbling was first used in [44] to study flowcharts and recursive schemata, and different variants of the game have later been applied to a rich selection of problems in computer science, including register allocation [50], algorithmic time and space trade-offs [17], parallel time [23], communication complexity [47], monotone space complexity [16, 28], cryptography [3, 22], and proof complexity [9, 11, 15] (where it should be emphasized that the above list of references is far from exhaustive). An excellent overview of pebbling up to ca 1980 is given in [46] and another in-depth treatment of some pebbling-related questions can be found in chapter 10 of [48]. Some more recent developments are discussed in the upcoming survey [42].

Let us briefly discuss what is known about space in proof complexity, since this is one of the two topics we are focusing on in this paper. The study of space in proof complexity was initiated in [25], which introduced the *clause space* measure for the well-known resolution proof system, a measure that has subsequently been thoroughly investigated. Informally, the clause space of a resolution proof can be defined as the maximal number of additional clauses – on top of the clauses in the original CNF formula – that a verifier needs to keep in memory at any time while checking the correctness of the proof.[1] While some formulas have proofs requiring only a small, sometimes even just constant, space overhead during verification, other formulas require a linear amount of extra space [1, 8, 25], and as shown in [25] no formulas require more than linear clause space in resolution.

Other papers have studied how space relates to other proof complexity measures. With respect to proof length, which can be viewed as a measure of (nondeterministic) running time, there is a wide range of trade-off results. It has been shown that there are formulas which have both short and space-efficient proofs, but as one of these measures is optimized the other one can blow up to almost worst-case behaviour [10]. Not only this, but there are even formulas where short proofs require more than the worst-case linear space [6, 7]. Yet other papers have studied other space measures such as *total space* [1, 12, 14], measuring the total number of symbols in a proof, and space complexity has also been considered for other proof systems than resolution. We refer the reader to the survey [41] for more details (although, for obvious reasons, it fails to cover the very latest results on total space).

All the space measures discussed above have in common the fact that they refer to the maximum space used at some point in the proof, but they are far from providing a complete picture of space usage during the whole proof. If we only know that a formula has high space complexity, it is not possible to distinguish between a formula that requires large space only at the beginning of the proof, say, and another that requires large space throughout the whole proof. This distinction might not be so important if we are considering the memory requirements of a verifier, since in this case we are chiefly interested in the

[1] Though slightly different from the definition in [25], this is equivalent up to a small additive constant.

maximum. However, it could be relevant for proof search: an algorithm that searches for a proof by producing clauses needs to discard many of them or risk exhausting its available memory. In this case, the difference between needing large space once versus at all times is the difference between making one lucky choice of which clauses to keep in memory versus being lucky all the time.

A similar issue occurs with so-called *memory-hard functions* in the context of cryptography. The idea behind memory-hard functions is that they should require a large amount of memory to evaluate, so that in order to compute such a function for many inputs as a part of a brute-force attack either an infeasible amount of memory is needed or the attack needs to be carried out sequentially. Yet, if the function only requires a large amount of memory during a limited time of the computation, then it is possible to reuse memory for different computations overlapping suitably in time as observed in [2]. Therefore, a more appropriate measure to analyse memory-hard functions is *cumulative space complexity* as introduced in [3], where one measures not the maximum memory consumption but the total memory usage aggregated over the time of the computation.

Although with hindsight this cumulative space complexity measure appears to be a very natural way of quantifying memory usage, it does not seem to have received too much attention in computational complexity theory, and to the best of our knowledge it has not been considered at all in the context of proof complexity. One of the main contributions of this paper is to transfer the concept of cumulative space to proof complexity and to initiate a study of this complexity measure for the resolution proof system.

Pebble games turn out to be a useful tool also for analysing cumulative space. For pebbling strategies cumulative space is straightforwardly defined as the sum over all steps of the pebbling of the number of pebbles on the DAG at each point in time. Thus, in the standard pebble game discussed above any DAG with n vertices can trivially be pebbled in time n and cumulative space $\mathrm{O}(n^2)$ by placing pebbles on all vertices in topological order. Since every vertex needs to be pebbled at some point, a trivial lower bound for the cumulative space is n. However, depending on the intended application one needs to consider other variations of this pebble game as discussed next.

In a proof complexity setting we need to study the *black-white pebble game*, which was introduced in [20] with the objective of modelling nondeterministic computations. Here white pebbles, corresponding to nondeterministic guesses, can be placed at any vertex at any time, but such a white pebble can only be removed from a vertex when all direct predecessors have (black or white) pebbles, corresponding to that the correctness of the nondeterministic guess can be verified.

To model parallel computation in a cryptographic setting, [3] introduced yet another pebble game, namely the *parallel (black) pebble game*. In this game, all the pebbling moves that are legal at some point in time can be performed simultaneously in one single step. This change of rules does not affect the maximal space required to pebble a DAG, but typically changes the pebbling time. Any connected DAG with a single sink requires linear time to pebble sequentially, but for a parallel pebbling it is easy to see that the time required is upper-bounded by the depth of the graph (i.e., the length of a longest path). We remark that an attractive feature of parallel pebbling is that it better captures the difference between maximal and cumulative space. Note that in any sequential pebbling game placing s pebbles requires s time steps, and during the last $s/2$ steps there will be at least $s/2$ pebbles on the DAG. Thus, any pebbling in maximal space s requires cumulative space $\Omega(s^2)$. In contrast, in a parallel pebbling the cumulative space can be small even when the maximal space is large.

1.1 Our Pebbling Contributions

In this paper, we study the cumulative space measure in the context of black-white pebbling. In order to do so, we also extend black-white pebbling to a parallel version. As pebble games go, this is a very powerful model, since it turns out that any DAG can be pebbled with a parallel black-white pebbling in constant time and linear cumulative space. Perhaps somewhat surprisingly, however, it is still possible to prove nontrivial time-space trade-offs. It can be shown that the parallel and sequential versions of black-white pebbling are closely connected (as discussed in more detail later in the paper), and therefore in this overview the exposition is focused on sequential black-white pebbling.

The first question we address is how the large cumulative space can be in the worst case for sequential black-white pebbling. As noted above, a trivial (black-only) pebbling in linear time and space has cumulative space $\mathrm{O}(n^2)$ for any graph over n vertices. In the other direction, the $\Omega(n/\log n)$ space lower bound in [30] already gives a $\Omega(n^2/\log^2 n)$ cumulative space lower bound for sequential black-white pebbling, as explained above. One cannot get a better cumulative space lower bound by this simple argument from maximal space lower bounds, however, since any DAG of constant indegree can be pebbled in maximal space $\mathrm{O}(n/\log n)$ [33].

We prove that the family of *grate graphs* in [49] require $\Omega(n^2)$ cumulative space for sequential black-white pebbling. This shows that for cumulative space it is not possible to improve on the trivial quadratic upper bound, in contrast to the maximal space measure where it is always possible to save a logarithmic factor from the trivial linear upper bound. This is also different from the parallel black pebble game, where there is a $\mathrm{o}(n^2)$ worst-case upper bound for cumulative space [2] and the best known cumulative space lower bound is $\Omega(n^2/\log n)$ [4]. In fact, it turns out that the difference between the sequential black-white and parallel black pebble games can be very large. We also prove that (a modified version of) the *butterfly graphs* in [51] require cumulative space $\Omega(n^2/\log n)$ in the sequential black-white pebble game but can be pebbled in linear cumulative space in the parallel black pebble game. Butterfly graphs also show that graphs that require large cumulative space do not necessarily require large maximal space, as they have logarithmic depth and thus can be pebbled in logarithmic space as observed in [33]. We obtain these results by studying the lower bounds on cumulative space in parallel black pebbling in [4] in terms of *depth-robustness* of graphs, and extending these lower bounds to other pebble games and other families of graphs.

Our next set of results concern trade-offs between time and space. Here our starting point is the family of *bit-reversal permutation graphs* studied in [37] which can be pebbled either with 3 pebbles or (as any graph) in linear time, but for which any pebbling in time t and space s must satisfy $t = \Omega(n^2/s^2)$, where as before n is the number of vertices in the graph.

We strengthen this trade-off to cumulative space, proving that pebblings of these graphs in space s require cumulative space $\Omega(n^2/s)$, which in particular implies that a pebbling in time $\mathrm{O}(n^2/s^2)$ must use space $\Omega(s)$ not only at some point but most of the time.[2] Furthermore, we establish an unconditional $\Omega(n^{3/2})$ cumulative space lower bound, which provides another example of graphs that require (at least somewhat) large cumulative space but can be pebbled in very small (even constant) maximal space. Our proofs of these results work by adapting the *dispersion* technique from [4]. This technique has the advantage that it isolates an abstract combinatorial property of the graph that makes the lower bound argument go through, and this cleaner approach enables us to prove these results not only

[2] Note, importantly, that such a space lower bound is *not* implied by the simple "space s implies cumulative space $\Omega(s^2)$" argument discussed previously.

for bit-reversal graphs but also for *random permutation graphs* (by showing that these graphs possess the required combinatorial property with high probability). To the best of our knowledge no trade-offs (even non-cumulative ones) were known for such graphs before for any flavour of the pebble game.

Finally, we consider a very concrete, extremal question regarding pebbling time-space trade-offs. It is an easy observation that any sequential black-white pebbling in constant space s can be carried out in time $\mathrm{O}(n^s)$, since there are only $\sum_{k=0}^{s} 2^k \binom{n}{k}$ possible different configurations of s pebbles in the graph, and no configuration repeats in a pebbling (or else the intermediate moves can be removed). In fact, a bit more thought reveals that this time bound can be sharpened to $\mathrm{O}(n^{s-1})$, since every configuration in space s is immediately followed by a pebble removal, and so we only need to consider distinct configurations of $s-1$ pebbles. It is a natural question whether this simple counting argument is in fact tight, so that there are graphs that can be pebbled in space s but where any such pebbling requires time $\Omega(n^{s-1})$.

For pebbling space $s = 3$, the minimum space in which any nontrivial pebbling strategy is possible, the bit-reversal graphs in [37] discussed above show that the answer to this question is affirmative. It is not hard to see that by stacking $s-2$ bit-reversal DAGs on top of one another, identifying the top layer in one graph with the bottom layer in the graph above, one obtains graphs that are pebblable in space s but where the obvious pebbling strategy achieving this bound requires time $\mathrm{O}(n^{s-1})$. We prove that this trivial upper bound is indeed asymptotically tight for any constant s.

1.2 Our Proof Complexity Contributions

Turning now to proof complexity, we consider the main contribution of our paper to be that we initiate the study of the cumulative space measure. While the concept of cumulative space seems to be as natural as maximal space, we are not aware of it having been studied in the context of proof complexity before. As was the case for the first papers on (maximal) space complexity in resolution [25], in this first paper on cumulative space in proof complexity we focus on the resolution proof system.

An immediate observation is that proof length is always a lower bound on cumulative space, and so exponential lower bounds on proof length – as shown for resolution in [18, 32, 52] and many later papers – trivially imply exponential lower bounds on cumulative space. Therefore, it seems that the cumulative space measure will be of independent interest mostly for formulas which have reasonably short proofs. An obvious candidate family to study are pebbling formulas [11], which have proofs in linear length, but which exhibit a rich variety of properties with respect to space complexity depending on the underlying graphs in terms of which they are defined.

However, we also need to decide on an appropriate model of the resolution proof system in which to study cumulative space. In the context of pebbling we concluded that cumulative space makes most sense for parallel versions of the pebble games, and so it is natural to ask whether one should consider a parallel version of resolution when studying cumulative clause space. It is not hard to argue that such a parallel model of resolution could be interesting in its own right, since it might be useful as a tool to analyse attempts to parallelize state-of-the-art SAT solvers using so-called *conflict-driven clause learning (CDCL)* [5, 38].

We define and study several different versions of the resolution proof systems with varying degrees of parallelity. The running time of parallel CDCL solvers has previously been analysed using resolution depth and the related *conflict resolution depth* and *schedule makespan* measures introduced in [36], and our models of parallel resolution allow us to reason about space in addition to time.

Similarly to what is the case for pebble games, our most general model of parallel resolution, where clauses can be inferred not just by syntactic application of the resolution rule but by semantic inference, is extremely powerful, so much so that it can deal with any formula in a constant number of steps and linear space. Since we can establish a tight relation between space and parallel speedup also for resolution, however, we can still obtain lower bounds when the maximal space is limited.

Studying pebbling formulas in these different models of resolution, and revisiting the reductions between resolution and pebble games in [9, 10], we can translate the pebbling results in Section 1.1 to results for the resolution proof system. Summarizing very briefly, we exhibit different formulas that have

- proofs in linear length but require quadratic cumulative space,
- proofs in logarithmic space but require $\Omega(n^2/\log n)$ cumulative space, and
- trade-offs between proof length and cumulative space.

1.3 Paper Outline

The rest of this paper is organized as follows. In Section 2 we present a more detailed overview of our pebbling results, introducing formal definitions of the pebble games and measures discussed above, and we give an analogous overview for resolution in Section 3. The reader is referred to the upcoming full-length version for all missing proofs. We conclude in Section 4 with a discussion of possible directions for future research.

2 Pebbling Results Overview

Let us start our pebbling overview by giving formal definitions of the basic concepts.

2.1 Definition of Pebble Games and Basic Properties

We say that a directed acyclic graph (DAG) $G = (V, E)$ with $|V| = n$ has *size* n. A vertex $v \in V$ has *indegree* δ if it has δ incoming edges $\{(u_1, v), \ldots, (u_\delta, v)\} \subseteq E$, $u_i \neq u_j$ for $i \neq j$, and we say that G has indegree δ if the maximum indegree of any vertex of G is δ. A vertex with no incoming edges is called a *source* and a vertex with no outgoing edges is called a *sink*. We say that a vertex u is a *predecessor* of a vertex v if there exists a directed path from u to v; moreover, if this path consists of only one edge then u is a *direct predecessor* of v. We denote by $\mathsf{parents}(v)$ the set of all direct predecessors of v. For technical reasons, it will sometimes be convenient to allow paths of length 0 in the definition above, so that a vertex can be a predecessor of itself. We will sometimes consider graphs obtained from other graphs by removing subsets of vertices, and for $U \subseteq V$ we write $G - U = \big(V \setminus U, E \setminus ((U \times V) \cup (V \times U)\big)$ to denote the DAG obtained from G by removing the vertices in U and all edges incident to U.

To get a unified description of all flavours of the pebble game discussed in Section 1, it is convenient to define pebbling as follows.

▶ **Definition 1** (Pebble games). Let $G = (V, E)$ be a DAG with a unique sink vertex z. The black-white pebble game on G is the following one-player game. At any time i, we have a *black-white pebbling configuration* $\mathbb{P}_i = (B_i, W_i)$ of black pebbles B_i and white pebbles W_i on the vertices of G, at most one pebble per vertex. The rules of how a pebble configuration $\mathbb{P}_{i-1} = (B_{i-1}, W_{i-1})$ can be changed to $\mathbb{P}_i = (B_i, W_i)$ are as follows:

1. A black pebble may be placed on a vertex v only if all immediate predecessors of v are covered by pebbles in both $\mathbb{P}_{i-1}$ and $\mathbb{P}_i$, i.e.,

$$v \in (B_i \setminus B_{i-1}) \;\Rightarrow\; \mathsf{parents}(v) \subseteq \mathbb{P}_{i-1} \cap \mathbb{P}_i \;.$$

   Note that, in particular, a black pebble can always be placed on a source vertex.
2. A black pebble on any vertex v in $\mathbb{P}_{i-1}$ can be removed in $\mathbb{P}_i$.
3. A white pebble can be placed on any vertex v in $\mathbb{P}_i$.
4. A white pebble on a vertex v in $\mathbb{P}_{i-1}$ may be removed in $\mathbb{P}_i$ only if all immediate predecessors of v are covered by pebbles in both $\mathbb{P}_{i-1}$ and $\mathbb{P}_i$, i.e.,

$$v \in (W_{i-1} \setminus W_i) \;\Rightarrow\; \mathsf{parents}(v) \subseteq \mathbb{P}_{i-1} \cap \mathbb{P}_i \;.$$

   In particular, a white pebble can always be removed from a source vertex.

A *legal pebbling* $\mathcal{P}$ of G is a sequence $\mathcal{P} = (\mathbb{P}_0, \ldots, \mathbb{P}_t)$ where every configuration $\mathbb{P}_i$ can be obtained from $\mathbb{P}_{i-1}$ using the rules 1–4. A *complete pebbling* $\mathcal{P} = (\mathbb{P}_0, \ldots, \mathbb{P}_t)$ is a legal pebbling where $\mathbb{P}_0 = \mathbb{P}_t = (\emptyset, \emptyset)$ and $z \in \bigcup_{i=0}^{t}(B_i \cup W_i)$ (i.e., the sink is pebbled at some point).

A *black pebbling* is a pebbling where $W_i = \emptyset$ for all $i \in [t]$. A pebbling is *sequential* if only a single application of a single rule 1–4 is used to get from from $\mathbb{P}_{i-1}$ to $\mathbb{P}_i$ for all $i \in [t]$. In a *(fully) parallel* pebbling an arbitrary number of applications of the rules 1–4 can be made to $\mathbb{P}_{i-1}$ to obtain $\mathbb{P}_i$ (but note that all pebble placements and removals have to be legal with respect to $\mathbb{P}_{i-1}$, and cannot make use of any pebble placements or removals made in parallel). Finally, we will also consider *parallel-black sequential-white* pebblings, which allows parallel applications of black pebble rules 1–2 to $\mathbb{P}_{i-1}$ to obtain $\mathbb{P}_i$, but only a single application of the white pebble rules 3–4. Note that, in the parallel setting, a simultaneous application of rules 1 and 4 on a same vertex replaces a white pebble by a black one.

The *time* of a pebbling $\mathcal{P} = (\mathbb{P}_0, \ldots, \mathbb{P}_t)$ is $t(\mathcal{P}) = t$; the (maximal) *space* is $s(\mathcal{P}) = s = \max_{i \in [t]} |B_i| + |W_i|$; and the *cumulative space* is $c(\mathcal{P}) = c = \sum_{i \in [t]} |B_i| + |W_i|$ (where we observe that $c \leq st$).

Parallel black pebbling was introduced in [3], where it was pointed out that for certain graphs parallel pebblings can be much more efficient than sequential, while for others they cannot do any better. For example, if we are considering time-space tradeoffs, any sequential black pebbling in space s and time t of the bit-reversal graph must satisfy $st = \Omega(n^2)$ [37], while in the parallel black game one can pebble such graphs in linear time and space $\mathrm{O}(\sqrt{n})$ [3]. In contrast, it was shown in [4] that there are graphs that can be pebbled sequentially in space s and time t satisfying $st = \mathrm{O}(n^2/\log n)$, but where these graphs even in the parallel model require not only $st = \Omega(n^2/\log n)$ but also cumulative space $\Omega(n^2/\log n)$.

Unlike the case of the black pebble game, we show that time and space in the black-white sequential and parallel games are closely related. Up to constant factors, it holds that if a parallel black-white pebbling $\mathcal{P}$ has maximal space s, then it is possible to save a factor s, but not more than a factor s, in time compared to a sequential black-white pebbling in the same space s.

▶ **Observation 2.** *Let $\mathcal{P}$ be a parallel black-white pebbling of a DAG G in time t, space s, and cumulative space c. Then there is a sequential black-white pebbling of G in time $2ts$, space $2s$, and cumulative space cs.*

Proof. Each parallel move places at most s pebbles and removes at most s pebbles, therefore we can simulate it by $2s$ sequential moves (making the pebble placements first, to make sure that these moves remain legal). ◀

▶ **Lemma 3.** *Let $\mathcal{P}$ be a sequential black-white pebbling of G in time t, space s, and cumulative space c, and let k be a positive integer. Then there is a parallel black-white pebbling of G in time $3\lceil t/k\rceil$, space $s+\lceil k/2\rceil$, and cumulative space $3\lceil c/k\rceil+t$.*

Proof. We divide $\mathcal{P}$ into $\lceil t/k\rceil$ intervals of (at most) k moves. We reorder the pebbling moves within each of these intervals so that we do all placements first and removals afterwards. This is still essentially a valid pebbling, because each configuration is a superset of the corresponding configuration in $\mathcal{P}$, except that we can possibly have vertices temporarily covered by several pebbles. The space usage in any intermediate configuration increases to at most $s+\lceil k/2\rceil$. We then collapse each subsequence into one parallel placement of white pebbles, one step replacing white pebbles with black pebbles as needed, and one parallel removal of black pebbles (this allows us to make all black pebble placements in parallel even though later black pebbles might be dependent on earlier pebble placements in the sequential pebbling). This decreases the time to $3\lceil t/k\rceil$.

To bound the cumulative space, note that if there is a configuration $\mathbb{P}_i$ in a sequential interval that has space s_i, then the corresponding three parallel configurations have aggregate space at most $3s_i+2x_j$, where x_j is the number of placements in that interval. Now consider a partition of the sequential pebbling into k subsequences $\mathbb{P}_i,\mathbb{P}_{i+k},\ldots,\mathbb{P}_{i+k(\lceil t/k\rceil-1)}$ of t/k configurations, evenly spaced, starting at $i\in[1,k]$. By an averaging argument, at least one of these k subsequences has a cumulative space of at most $\lfloor c/k\rfloor$. Hence the total cumulative space is at most $\sum_{j\in[t/k-1]}(3s_{i+j}+2x_{i+j})=3\sum_{j\in[t/k-1]}s_{i+j}+2\sum_{j\in[t/k-1]}x_j\geq 3c/k+2t/2$. ◀

Observe that when $k=\Theta(s)$ the cumulative space in Lemma 3 is dominated by the term t, so we only save a factor s in cumulative space when the sequential pebbling has cumulative space $c=\Theta(st)$. Since the graphs we will discuss in what follows have cumulative space lower bounds of this form, studying the sequential game already gives us all the information we want about the parallel game.

▶ **Corollary 4.** *Let $\mathcal{P}$ be a black-white pebbling of G in time t and space s. Then there is a parallel black-white pebbling of G in time $\lceil t/2s\rceil$, space $4s$, and cumulative space $2t$.*

2.2 Robustness and High Cumulative Space Complexity

We proceed to define the concept of *depth-robustness* of graphs, which is inspired by [24, 45] and which will be central to our work.

▶ **Definition 5** ($\mathcal{G}$ -robustness)**.** Let $\mathcal{G}$ be a family of DAGs and let $e,d\in\mathbb{N}^+$ be positive integers. We say that a DAG $G=(V,E)$ is (e,d)-*$\mathcal{G}$-robust* if for every subset of vertices $U\subseteq V$ of size at most e it holds that $G-U$ contains a subgraph $H\in\mathcal{G}$ of size at least d.

When $\mathcal{G}$ is the class of directed paths, then we say that G is *depth-robust*, and when $\mathcal{G}$ is the class of DAGs with one sink the DAG G is said to be *predecessor-robust*.[3]

For our pebbling lower bounds we are interested in graphs with very high robustness, i.e., for as large values of e and d as possible. Depth-robustness was first studied by Erdős, Graham and Szemerédi [24] who showed how to construct DAGs with indegree $\Theta(\log(n))$ possessing $(\Omega(n),\Omega(n))$-depth-robustness. However, in our applications it is important that

[3] This choice of terminology is inspired by [45], which discusses the dual notions of “depth-separators” and “predecessor-separators.”

the graphs have *constant* indegree. Valiant [53] showed that for constant indegree and linear depth the best we can hope for is $(\mathrm{O}(n/\log n), \mathrm{O}(n))$-depth-robustness. Fortunately for us, it was shown in [4, 45] that such extremal $(\Theta(n/\log n), \Theta(n))$-depth-robust graphs do exist. Conversely, if we want constant indegree with the parameter e linear in the graph size, then $(\epsilon n, n^{1-\epsilon})$-depth-robustness is the best we can hope for [53]. In [49] a family of constant-indegree $(\Theta(n), \Theta(n^{1-\epsilon}))$-depth-robust graphs were presented.

The connection between depth-robustness and cumulative space was made in [4], where it was shown that an (e, d)-depth-robustness graph requires parallel black cumulative space at least ed. In this work, we give a more general theorem of this form for the case of $\mathcal{G}$-robustness. We then use this theorem to obtain the following lower bounds for depth-robust and predecessor-robust graphs.

▶ **Lemma 6.** *If G is an (e, d)-depth-robust DAG, then G requires sequential black-white cumulative space at least ed, and parallel-black sequential-white cumulative space at least $e\sqrt{d}$.*

▶ **Lemma 7.** *If G is an (e, d)-predecessor-robust DAG, then G requires black-white cumulative space at least ed.*

Focusing on the range of parameters discussed above, we can see that, it follows from Lemmas 6 and 7 that a $(\Theta(n/\log n), \Theta(n))$-depth-robust graph has sequential black-white cumulative space complexity $\Omega(n^2/\log n)$ and parallel-black sequential-white pebbling space complexity $\Omega(n^{3/2}/\log n)$.

A class of DAGs that are predecessor-robust are *grates* – graphs with n' sources and n' sinks such that after the removal of an arbitrary set of kn' vertices (for some constant k) there are still a linear number of sources and sinks that are all pairwise connected. *Butterfly graphs* [51] are grates with $n = n' \log n'$ vertices that are $(\Theta(n/\log n), \Theta(n/\log n))$-predecessor-robust. Moreover, it is not hard to show that if we append n' single-sink DAGs of size $\log n'$, one to each source of the butterfly graph, the resulting graph is $(\Theta(n/\log n), \Theta(n))$-predecessor-robust. This implies that these graphs require cumulative space $\Omega(n^2/\log n)$. Note that butterfly graphs (also in the modified version just described) can be pebbled with $\mathrm{O}(\log n)$ pebbles (since the graphs have depth $\mathrm{O}(\log n)$), and thus it is not the case that high cumulative space implies large maximal space.

It has been established that extremal depth-robustness is both a necessary [2] and sufficient [4] condition to have high cumulative space in the parallel black game. In particular, using the fact that no graph of size n with constant indegree is $(\omega(n/\log n), \Theta(n))$-depth-robust, it was shown in [2] that in the parallel black pebble game, for any constant $\epsilon > 0$, any such graph has cumulative space complexity $\mathrm{o}(n^2/\log^{1-\epsilon} n)$. A natural question is if this also holds for black-white pebbling. We show that this is not the case: there are graphs that have maximum cumulative space complexity $\Omega(n^2)$ in the black-white pebble game. This follows from Lemma 7 and the existence of grates of size linear in the number of sources and sinks [49].

2.3 Dispersion and Cumulative Space Trade-Offs

Another property of graphs that is important in the current paper is *dispersion*. This notion was used in [4] to obtain another condition ensuring high parallel black cumulative space complexity. We define two similar concepts and then use them to obtain cumulative space trade-offs. The results we get are for two classes of *permutation graphs* – graphs that consist two ordered paths of vertices $1, 2, \ldots, n$, where in addition an edge is added from each vertex i in the first path to its image under some specified permutation σ in the second path.

A family of permutations that will be of particular interest to us are the so-called *bit-reversal permutations*, which are defined for $n = 2^m$ and which simply reverse the binary representations of numbers. That is, if $j = (b_1 \cdots b_m)_{(2)}$, then the bit-reversal permutation σ sends j to $\sigma(j) = (b_m \cdots b_1)_{(2)}$. It was previously known [37] that any sequential black-white pebbling of a bit-reversal permutation graph on $2n$ vertices in time t and space s satisfies $st = \Omega\big(n^2/s\big)$. Moreover, it was shown in [37] that this is tight up to constant factors and that there is a black-white pebbling in time t and space s such that $st = \mathrm{O}\big(n^{3/2}\big)$.

We observe that while bit-reversal graphs are *not* $(2\sqrt{n}, 2\sqrt{n})$-depth-robust, they can be shown to be $(\sqrt{n}, n)$-predecessor-robust. Therefore, in constrast to [4], where it was not possible to establish a parallel black cumulative space lower bound of $n^{3/2}$ using depth-robustness, we are able to obtain a black-white cumulative space lower bound of $n^{3/2}$ using predecessor-robustness.

Our reason for studying dispersion properties of bit-reversal graphs is to characterize how cumulative space increases when space decreases. We show that the time-space trade-off in [37] can be strengthened to a cumulative space trade-off. Our result implies that if $\mathcal{P}$ is a sequential black-white pebbling of a bit-reversal graph in space s and time n^2/s^2, then it needs to use space s not only at some point of the pebbling, but during a large part of the time.

An advantage of our approach is that we identify a general property of graphs that imply cumulative space trade-offs, so that the task of establishing a trade-off reduces to proving that the graph has this desired property. As a consequence of this simplification, we are able to prove the same kind of trade-off results not only for bit-reversal graphs but also for random permutation graphs, a class of graphs for which it seems nothing was known before.

▶ **Theorem 8.** *If G is a random permutation graph, then it holds asymptotically almost surely that in the sequential black-white pebble game G requires cumulative space $\Omega(n^{3/2})$ and any pebbling $\mathcal{P}$ of G in maximal space s has cumulative space $\Omega(n^2/s)$.*

2.4 Pebblings in Small Space Can Require Maximum Length

Let us finally consider the question of how long a shortest sequential pebbling of a graph can be given constraints on the maximal pebbling space. Without loss of generality, a black pebbling in space s takes time at most $\binom{n}{\leq s} \leq n^s$, simply because there is no need to repeat any pebble configuration. A moment of thought reveals that in fact we get the upper bound $\binom{n}{s-1} + \binom{n}{\leq s-1} \leq n^{s-1}$, since every configuration in maximal space s is followed by an erasure yielding a space-$(s-1)$ configuration, and these configurations also do not repeat. For black-white pebbling the upper bound becomes $2^{s-1}\big(\binom{n}{s-1} + \binom{n}{\leq s-1}\big) \leq 2^{s-1}n^{s-1}$.

As discussed in the introduction, it can be read off from [37] that for space-3 pebblings the $\mathrm{O}\big(n^2\big)$ upper bound is tight up to constant factors – bit-reversal DAGs are examples of graphs for which pebblings in optimal space 3, or indeed any constant space, require quadratic time. We extend this result to any $s = \mathrm{O}(1)$ by exhibiting graphs that can be pebbled in space s but where any such pebbling requires time $\Omega\big(n^{s-1}\big)$. We do this by generalizing permutation graphs to multiple layers, where we have k directed path graphs of length n and $k-1$ layers of permutations between the vertices $1, 2, \ldots, n$ in consecutive paths (so that the permutation graphs considered in [37] are 2-layer bit-reversal graphs with paths of length n). We state two theorems below for the black and black-white sequential pebble games, and just as for the 2-layer graphs in [37] our bounds can be stated not just for minimal space but also an arbitrary space parameter s greater than this minimum.

▶ **Theorem 9.** *Let G be a k-layer bit-reversal graph with paths of length n. Then for any s such that $k+1 \leq s \leq \sqrt{n}$ there exists a sequential black pebbling of G in space s and time $\mathrm{O}(n^k/s^{2k-3})$. Furthermore, every sequential black pebbling of G in space s requires time $\Omega(n^k/s^{2k-3})$.*

▶ **Theorem 10.** *Let G be a k-layer bit-reversal graph with paths of length n. Then for any s such that $k+1 \leq s \leq \sqrt{n}$ there exists a sequential black-white pebbling of G in space s and time $\mathrm{O}(n^k/s^{2k-2})$. Furthermore, every sequential black-white pebbling of G in space s, requires time $\Omega(n^k/s^{2k-2})$.*

Our proofs of these results are inspired by the reasoning in [37] for 2-layer permutation graphs, but we also need to overcome some new challenges. The essence of the argument is that in order to place a pebble on the jth layer we need to do some work on the preceding layer. If we only have two layers the argument ends here, but when we want to apply the argument recursively we need to be more careful. Indeed, placing pebbles on the $(j-1)$st layer will now require placing more pebbles on the $(j-2)$nd layer, but if we choose the order in which we do the pebble placements wisely, we may be able to reuse part of the work in the $(j-2)$nd layer for several pebble placements in the $(j-1)$st layer. We are able to find a strategy to exploit this insight and obtain optimal upper bounds, but also to make the lower bound argument resilient enough to get asymptotically matching lower bounds.

3 Cumulative Space for the Resolution Proof System

We now proceed to describe in more detail the proof complexity results in our paper. We start this section by a brief review of some standard proof complexity preliminaries, after which we discuss how to refine the definition of the resolution proof system to be able to make meaningful and precise claims about maximal space and cumulative space. This then allows us to make the connection to the pebbling results in Section 2 and what proof complexity implications they have.

A *literal* over a Boolean variable x is either x itself (a *positive literal*) or its negation $\overline{x}$ (a *negative literal*). A *clause* $C = a_1 \vee \cdots \vee a_k$ is a disjunction of literals a_i over pairwise disjoint variables. A k*-clause* is a clause that contains at most k literals. A *CNF formula* $F = C_1 \wedge \cdots \wedge C_m$ is a conjunction of clauses and a k*-CNF formula* is a CNF formula consisting of k-clauses. We think of clauses and CNF formulas as sets: order is irrelevant and there are no repetitions.

The standard definition of a *resolution refutation* $\pi : F \vdash \bot$ of an unsatisfiable CNF formula F – or a *resolution proof* for (the unsatisfiability of) F – is as an ordered sequence of clauses $\pi = (D_1, \ldots, D_t)$ such that $D_t = \bot$ is the empty clause containing no literals, and each clause D_i, $i \in [t]$, is either an *axiom* $D_i \in F$ or is derived from clauses D_j and D_k, $j, k < i$, by the *resolution rule*

$$\frac{B \vee x \qquad C \vee \overline{x}}{B \vee C}\,, \tag{1}$$

where we refer to $B \vee C$ as the *resolvent over* x of $B \vee x$ and $C \vee \overline{x}$.

In order to study space in general, and cumulative space in particular, we refine the above definition into a family of proof systems as follows.

▶ **Definition 11** (Resolution)**.** A resolution refutation $\pi : F \vdash \bot$ of a CNF formula F is a sequence of *configurations*, or sets of clauses, $\pi = (\mathbb{C}_0, \ldots, \mathbb{C}_t)$ such that $\mathbb{C}_0 = \emptyset$, $\bot \in \mathbb{C}_t$, and for all $i \in [t]$ we obtain $\mathbb{C}_i$ from $\mathbb{C}_{i-1}$ by applying exactly one of the following type of rules:

Axiom download Add $A \in F$.
Inference Add D derived from clauses in $\mathbb{C}_{i-1}$.
Erasure Remove clauses from $\mathbb{C}_{i-1}$.
We say that a refutation is (a) *sequential* if at every time step we apply the chosen rule exactly once; (b) *inference-parallel* if only one clause can be downloaded but the inference rule can be applied an arbitrary number of times (but always deriving from $\mathbb{C}_{i-1}$); and (c) *fully parallel* (or just *parallel*) if both axiom download and inference rules can be applied an arbitrary number of times (but note that we cannot mix applications of different rules in the same step). Furthermore, a refutation is said to be (1) *syntactic* if inferences use the resolution rule (1) and (2) *semantic* if instead any clause D such that $\mathbb{C}_{i-1} \vDash D$ can be inferred immediately.

The *length* of a resolution refutation π is the number of derivation steps t and the *size* is the total number of clauses introduced in downloads and inference steps (counted with repetitions). The *maximal (clause) space*, or just *space*, of π is $\max\{|\mathbb{C}_i| : \mathbb{C}_i \in \pi\}$ and the *cumulative (clause) space* is $\sum_{\mathbb{C}_i \in \pi} |\mathbb{C}_i|$.

Note that Definition 11 yields a total of six different flavours of resolution 1(a)–2(c) depending on the amount of parallelism and on whether inferences are syntactic or semantic. In what follows, we will discuss our motivation for considering these different models and what we can say about them.

A first, general comment is that from a proof complexity point of view we are mainly interested in *syntactic* versions of the proof systems in Definition 11. Strictly speaking, the *semantic* versions are not even propositional proof system in the sense of Cook and Reckhow [19], since we do not know how to verify semantic implications in polynomial time. In any semantic system we can download all axioms in the formula and then derive contradiction in a single inference step, and efficiently verifying such an inference means solving SAT in polynomial time. However, most results on (clause) space in the proof complexity literature actually hold in the stronger semantic setting. For maximal space this is not so surprising, since the semantic and syntactic space measures are within a constant factor of each other [1], but even for trade-offs one tends to get results in the semantic setting for free (with the notable exceptions of [6, 7]).

Syntactic sequential resolution is the standard definition discussed at the beginning of this section (and note that for this version of resolution the length and size measures are essentially the same). A somewhat unsatisfactory feature of this model is that (analogously to what is the case for pebbling) a maximal space lower bound s immediately implies a cumulative space lower bound $\Omega(s^2)$. The reason is completely analogous: since we can only infer one new clause per time step, during the $s/2$ time steps before reaching space s we must have had at least $s/2$ clauses in memory. It turns out, however, that we can actually beat this lower bound in certain settings, and we also remark that cumulative length-space trade-offs do not necessarily follow from such trivial arguments and so make sense even for syntactic sequential resolution.

By allowing parallel application of inference steps we want to try to get away from cumulative space lower bounds that hold only for the trivial reason just discussed. In *syntactic inference-parallel resolution* we therefore allow clauses to be derived in parallel. As it turns out, anything we are currently able to prove for this model we can also establish for the stronger *semantic inference-parallel resolution* system.

We can also go in the other direction from the syntactic sequential model and introduce a parallelism of sorts by studying *semantic sequential resolution*. As already alluded to, this is a very powerful system since any formula can be refuted in linear size and space by downloading all its axioms in a linear number of steps and then deriving contradiction in just one semantic inference step, but nevertheless the space lower bounds and length-space trade-offs in [9, 10] hold in this model, and can in fact be verified to hold even for semantic inference-parallel resolution.

The most challenging models in terms of lower bounds are the fully parallel ones. *Syntactic parallel resolution* could be viewed as a potentially interesting model for proving lower bounds on parallel SAT solvers using conflict-driven clause learning, where one could imagine an arbitrarily large number of solvers producing resolvents in parallel and having perfect access to shared memory. It is not hard to see that if a standard resolution proof is represented as a DAG in the natural way, then syntactic parallel length, which would be a proxy for execution time, is just the depth of this DAG.

In the semantic model, adding also parallel axiom downloads makes the proof system exceptionally powerful, since now any formula can be refuted in constant length 2, linear size, and linear cumulative space. This seems a bit too strong to be really interesting (and can be viewed as a reason for preferring the inference-parallel version described previously). However, we shall see that even for semantic fully parallel resolution it is still possible to obtain nontrivial trade-off results if the maximal (non-cumulative) space is bounded.

Moving on from this philosophical discourse to a more concrete discussion of results, we note that most of the proof complexity consequences we derive from the pebbling results in Section 2 are for semantic inference-parallel resolution, and thus hold for all models above except the fully parallel ones. We start by reporting a disappointing fact, however: even in semantic inference-parallel resolution we have the problem that cumulative space is at least maximal space squared.

▶ **Lemma 12.** *If F requires maximal space s in semantic inference-parallel resolution, then any semantic inference-parallel refutation of F has cumulative space $\Omega(s^2)$.*

Proof. For simplicity let us think of each step in a semantic inference-parallel resolution refutation as being either an inference-plus-erasure step or a download step. Clearly, this can only affect the clause space measure by a factor 2.

An inference-plus-erasure step can be seen as a compression operation. Since the proof system is semantic, we only care about the information contained in a configuration, and since an inference step cannot increase the information but only add explicitly clauses that are already implied by the configuration, there is no need to add any extra clauses on top of the minimum amount needed to encode the semantic information we want the proof to maintain at this point. Therefore, without loss of generality the number of clauses only increases at download steps, and since these are sequential we can conclude that the number of clauses increases by at most 1 at every step.

But this means that we can apply the same argument as for syntactic sequential resolution above: during the $s/2$ time steps preceding a space-s configuration we must have at least $s/2$ clauses in memory, and hence a cumulative lower bound $\Omega(s^2)$ follows. ◀

It is important to note, though, that Lemma 12 has no implications for cumulative space trade-offs for formulas where the maximal space complexity is at most $O(\sqrt{N})$ measured in the formula size N, since in this setting the max-space-squared argument only implies a trivial $\Omega(N)$ cumulative space lower bound, and we present such trade-off results below that do not follow from Lemma 12. We also report results that asymptotically beat the maximal-space-squared lower bound for cumulative space.

In order to obtain these results, we need to review how our cumulative pebbling results in Section 2 can be translated to claims about resolution refutations of so-called *XORified* *pebbling formulas.* We will be very brief here, since all that needs to be done is to read the pebbling-to-resolution reductions in [10] and verify that the proofs work not only for semantic sequential resolution but also for semantic inference-parallel resolution. We just state the reduction that we need below, since we can use it in a completely black-box fashion without knowing any details about what these formulas are. The interested reader is referred to [10] for the missing details.[4]

▶ **Theorem 13** (by the proof of Theorem 2.1 in [10]). *Let π be a semantic inference-parallel resolution refutation of a XORified pebbling formula $Peb_G[\oplus]$ in length L, maximal space s, and cumulative clause space c. Then there is a sequential black-white pebbling of the underlying DAG G in time L, space s, and cumulative space c.*

Analogously to what is the case in [10], the generic reduction in Theorem 13 can now be applied to a multitude of different graph families with different pebbling properties to yield CNF formulas with the same properties in resolution. Below we just give a sample of such results that we find particularly interesting.

For maximal space it is known that formulas refutable in linear size $\mathrm{O}(N)$ never require space more than $\mathrm{O}(N/\log N)$. For cumulative space the lower bound can be truly quadratic, however, beating the max-space-squared bound in Lemma 12 by a factor $\log^2 N$.

▶ **Theorem 14.** *There is a family of 6-CNF formulas $\{F_N\}_{N\in\mathbb{N}^+}$ of size $\Theta(N)$ that have syntactic sequential resolution refutations in size $\mathrm{O}(N)$, and hence also in maximal clause space $\mathrm{O}(N/\log N)$, but for which any semantic inference-parallel refutations require cumulative clause space $\Omega(N^2)$.*

This theorem follows from studying pebbling formulas defined in terms of *grate graphs* as in [49] and using that the high predecessor-robustness of these graphs imply strong lower bounds on cumulative space as stated in Section 2.

A natural question is what cumulative space tells us about maximal space, and in particular whether high cumulative space complexity implies that the maximal space complexity must also be large. This might sound intuitively plausible, but turns out to be false in a very strong sense.

▶ **Theorem 15.** *There is a family of 6-CNF formulas $\{F_N\}_{N\in\mathbb{N}^+}$ of size $\Theta(N)$ that can be refuted in syntactic sequential resolution in size $\mathrm{O}(N)$ and also in maximal clause space $\mathrm{O}(\log N)$, but for which any semantic inference-parallel refutations require cumulative clause space $\Omega(N^2/\log N)$.*

Here the graphs we need are surprisingly simple, namely butterfly graphs. They again have high predecessor-robustness, but since they are shallow the pebbling formulas generated from them have refutations in small maximal space.

Finally, we turn to the question of length-space trade-offs. We remark that in a cumulative space setting formulas for which small-space proofs require superpolynomial length, as in the strongest results in [10, 7, 6], are not too interesting, since length is trivially a lower bound on cumulative space. Rather, we focus on formulas for which small-space proofs incur only a

[4] It might be worth noting, though, that just as in [10] our results hold not only for pebbling formulas substituted with exclusive or – substitution with any so-called *non-authoritarian* (or *robust*) function that can never be fixed by restricting any single variable to some value works fine. Binary exclusive or is just the simplest example of such a function, whereas standard or is a simple non-example since setting a single variable to true fixes the value of the function to true.

polynomial blow-up in proof length. Can we find such formulas for which it holds not only that short proofs must have large maximal space s, but where such short proofs must be memory-intensive in that this amount of space s must be used essentially throughout the whole proof? The answer to this question is yes, and one example are pebbling formulas over the bitreversal permutation graphs studied in [37]. The next theorem follows by combining the reduction in Theorem 13 with the fact that bitreversal graphs are dispersed as stated in Section 2.

▶ **Theorem 16.** *There is a family of* 6*-CNF formulas* $\{F_N\}_{N\in\mathbb{N}^+}$ *of size* $\Theta(N)$ *such that for any* $s = \mathrm{O}(\sqrt{N})$ *the formula* F_N *has a syntactic sequential resolution refutation in size* $\mathrm{O}(N^2/s^2)$ *and maximal clause space* $\mathrm{O}(s)$*, but any semantic inference-parallel refutation of* F_N *in maximal space* s *requires cumulative clause space* $\Omega(N^2/s)$*.*

In particular, a proof in maximal space s has length $\Omega(N^2/s^2)$, and if furthermore the proof has length $\mathrm{O}(N^2/s^2)$, then $\Omega(N^2/s^2)$ of the configurations have space $\Omega(s)$. Hence, these formulas have syntactic sequential resolution refutations in simultaneous length $\mathrm{O}(N)$ and space $\mathrm{O}(\sqrt{N})$, but any semantic inference-parallel refutation with the same parameters has $\Omega(N)$ configurations with space $\Omega(\sqrt{N})$. We remark that this result makes sense even in the weaker syntactic sequential model, since maximal space $\Omega(\sqrt{N})$ only implies a trivial $\Omega(N)$ cumulative space lower bound.

As already noted, semantic fully parallel resolution is an extremely powerful model, since we can refute any formula with just one (parallel) axiom download step followed by one (semantic) inference step, but if we limit the available space then the usefulness of parallelism is restricted. In fact, the speed-up from parallelism is proportional to the space.

▶ **Observation 17.** *Let* π *be a semantic parallel resolution refutation of a formula* F *in length* L*, maximal clause space* s*, and cumulative clause space* c*. Then there is a semantic sequential refutation of* F *in length* Ls*, maximal clause space* s*, and cumulative clause space* cs*.*

Proof. Each parallel axiom download or inference adds at most s new clauses, therefore we can simulate it by s sequential axiom downloads or inferences respectively. ◀

Using Observation 17 we can transfer the trade-offs above from inference-parallel to fully parallel semantic resolution by sacrificing a factor s.

▶ **Lemma 18.** *Let* π *be a syntactic sequential resolution refutation of a formula* F *in length* L*, maximal space* s*, and cumulative space* c*, and let* $\ell \in \mathbb{N}^+$ *be a positive integer. Then there is a semantic parallel resolution refutation of* F *in length* $3\lceil L/\ell\rceil$*, maximal space* $s + \lceil \ell/2\rceil$*, and cumulative space* $3\lceil c/\ell\rceil + L$*.*

Proof sketch. Analogously to the proof of Lemma 3, we divide π into L/ℓ intervals of ℓ steps each. We reorder derivation steps within every interval so that we do all axiom downloads first, inferences next, and removals at the end of the interval. We then collapse each sequence into one axiom download, one inference, and one removal step. ◀

Let us finally just observe that although proving strong lower bounds for the fully parallel versions of resolution looks like a formidable challenge, which we leave as future work, we can obtain a simple separation between semantic and syntactic fully parallel resolution.

▶ **Proposition 19.** *Every syntactic, fully parallel resolution refutation of a minimally unsatisfiable CNF formula in space* $s \le N$ *requires length* $N/s + \log s - 2$*.*

Proof Sketch. Since the inference rule is binary, the number of useful clauses in the second-to-last-last configuration, namely those used to infer contradiction, is at most 2. Analogously, the number of useful clauses in the ith last configuration is at most 2^i. Hence, in the last $\log s$ steps we see at most $2s$ useful clauses in total. Since we need to see each axiom at least once, we still need at least $N/s - 2$ more steps. ◀

In particular, any syntactic refutation requires length $\log N$, and a refutation in this length requires space $\Omega(N)$. This is in contrast to semantic refutations, which have proofs in length 2, and no space lower bound other than the trivial N/L.

By way of example, consider a (plain) pebbling formula on a path graph of length N. A syntactic refutation in length $\log N$ requires space $\Omega(N)$, while there exists semantic refutation in length $\log N$ and space $2N/\log N + \mathrm{O}(1)$: just download the axioms corresponding to $2N/\log N$ consecutive vertices at a time and infer one new clause.

While this is technically a separation, it is also very brittle. For any integer k, it is possible to find a syntactic proof in length $(1+1/k)\log N$ and space $2kN/\log N+\mathrm{O}(1)$. First, download the axioms corresponding to evenly spaced vertices at distance $\log N/k$. Then for $\frac{2}{k}\log N$ steps download the clauses corresponding to the previous and next vertex and do a parallel inference step. Another inference step leaves us with a path of length $N/k\log N$, which we can trivially refute in length $\log N - \log k - \log\log N$ and space $N/k\log N$.

4 Concluding Remarks

In this paper, we study space complexity with a focus not on peak memory usage but on aggregated memory consumption over the whole computation. We consider two computational models, namely pebble games on DAGs and the resolution proof system in proof complexity.

For black-white pebbling, which is a model of nondeterministic computation, we prove optimal cumulative space lower bounds and also time-space trade-offs where in order to achieve optimal time the space needs to be large not only at a single point in time but throughout essentially the whole computation. We do so by studying the concepts of *depth-robustness* and *dispersion* of graphs, drawing on and extending work in [2, 3, 4] and other papers, and proving that different graph families of interest possess these properties.

In the context of proof complexity we are not aware of the cumulative space measure having been studied before, and so our first contribution here is to give a suitable formal definition, and also to consider different, more or less parallel, versions of the resolution proof system in which it makes sense to study cumulative space. We then use, and slightly extend, the reductions between pebbling and resolution in [9, 10] to transfer our lower bounds and trade-off results for pebbling also to resolution.

Since, to the best of our knowledge, ours is the first paper to study cumulative space both for black-white pebbling and for proof complexity, it is perhaps not so surprising that there is a wealth of open problems that this paper does not resolve. Below, we briefly discuss some possible directions for future research.

One set of questions on which we make progress but which we do not answer completely concern the relation between maximal space and cumulative space. For sequential black-white pebblings of n-vertex DAGs we prove an optimal $\Omega(n^2)$ cumulative space lower bound for a particular family of DAGs, but for graphs that can be pebbled in maximal space $\mathrm{O}(\log n)$ we only obtain a $\Omega(n^2/\log n)$ cumulative space lower bound and for graphs pebblable in space $\mathrm{O}(1)$ the best cumulative bound we can get is $\Omega(n^{3/2})$. Could it be the case that there are graphs that can be pebbled in maximal space $\mathrm{O}(1)$ but nevertheless require cumulative space $\Omega(n^2)$? Or do strong enough cumulative space lower bounds by necessity imply also nontrivial maximal space lower bounds?

It has been shown for parallel black pebbling that extremal depth-robustness is both necessary and sufficient for a graph to have high cumulative space complexity. We prove that for black-white pebbling predecessor-robustness is sufficient to imply high cumulative space, but leave open whether this condition is necessary or not.

For standard time-space trade-offs in sequential pebbling, it was shown in [37] that bit-reversal DAGs have a black pebbling trade-off of the form $t = \Theta(n^2/s)$ whereas for black-white pebbling the trade-off is a slightly weaker $t = \Theta(n^2/s^2)$. It was conjectured in [37] that there are other permutation graphs for which the black-white pebbling trade-off could also be shown to be an optimal $t = \Theta(n^2/s)$. One natural candidate class of graphs to consider in this context are graphs obtained from random permutations, and this is the original reason why we were interested to study them in this paper. So far we were only able to obtain trade-offs with the same parameters as for bit-reversal DAGs, but it is an interesting question whether our tools could be sharpened to prove even stronger trade-offs results for random permutation graphs.

Turning to our proof complexity results, they can be seen to be yet another contribution to the sequence of papers [39, 43, 9, 40, 10] obtaining space bounds and time-space trade-offs in proof complexity by instead studying pebble games and reductions between pebblings of DAGs and resolution refutations of so-called pebbling formulas defined in terms of these DAGs. While these connections have turned out to be very fruitful, it would also be interesting to go beyond pebbling formulas and explore whether cumulative space results could be obtained for, e.g., Tseitin formulas on long and narrow rectangular grids as studied in [6, 7] or for other formulas.

One motivation behind our models of parallel resolution was the connection to parallel SAT solving, but our models do not take into account practical limitations such as the number of computing nodes or the communication between nodes. Could there be natural ways to incorporate such limitations, and could this also provide a better understanding of parallel resolution?

Another, somewhat related, question is whether formulas possessing strong cumulative space lower bounds are hard also in practice for (sequential or parallel) SAT solvers. Just maximal space lower bounds do not seem to be sufficient to imply practical hardness, as shown, e.g., in the fairly extensive empirical experiments on pebbling formulas in [35], but perhaps cumulative space could be a more relevant concept in this context.

Finally, it can be noted that our study of cumulative space in proof complexity as initiated in this paper is limited to the resolution proof system. This is mostly because resolution is the proof system where space complexity is best understood, and where the toolbox for studying these questions is most well developed. However, different concepts of maximal space and time-space trade-offs have been studied also for other proof systems such as polynomial calculus [1, 7, 13, 26, 27] and cutting planes [21, 29, 31, 34], and it would be interesting to extend the study of cumulative space to these proof systems.

Acknowledgements. The authors wish to thank Yuval Filmus for making us aware of each other's existence and thus providing the stimulus for the joint work that led up to this paper. Different subsets of the authors are grateful to Ilario Bonacina, with whom we had stimulating discussions during various stages of this project and who, in particular, provided valuable insights on dispersion, and to Adam Schill Collberg and Jan Elffers for helpful discussions on random permutation graphs.

References

1 Michael Alekhnovich, Eli Ben-Sasson, Alexander A. Razborov, and Avi Wigderson. Space complexity in propositional calculus. *SIAM Journal on Computing*, 31(4):1184–1211, 2002. Preliminary version in *STOC '00*.

2 Joël Alwen and Jeremiah Blocki. Efficiently computing data-independent memory-hard functions. In *Proceedings of the 36th Annual International Cryptology Conference (CRYPTO '16)*, pages 241–271, August 2016.

3 Joël Alwen and Vladimir Serbinenko. High parallel complexity graphs and memory-hard functions. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing (STOC '15)*, pages 595–603, June 2015.

4 Joël Alwen, Jeremiah Blocki, and Krzysztof Pietrzak. Depth-robust graphs and their cumulative memory complexity. Technical Report 2016/875, Cryptology ePrint Archive, September 2016.

5 Roberto J. Bayardo Jr. and Robert Schrag. Using CSP look-back techniques to solve real-world SAT instances. In *Proceedings of the 14th National Conference on Artificial Intelligence (AAAI '97)*, pages 203–208, July 1997.

6 Paul Beame, Chris Beck, and Russell Impagliazzo. Time-space tradeoffs in resolution: Superpolynomial lower bounds for superlinear space. *SIAM Journal on Computing*, 45(4):1612–1645, August 2016. Preliminary version in *STOC '12*.

7 Chris Beck, Jakob Nordström, and Bangsheng Tang. Some trade-off results for polynomial calculus. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC '13)*, pages 813–822, May 2013.

8 Eli Ben-Sasson and Nicola Galesi. Space complexity of random formulae in resolution. *Random Structures and Algorithms*, 23(1):92–109, August 2003. Preliminary version in *CCC '01*.

9 Eli Ben-Sasson and Jakob Nordström. Short proofs may be spacious: An optimal separation of space and length in resolution. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS '08)*, pages 709–718, October 2008.

10 Eli Ben-Sasson and Jakob Nordström. Understanding space in proof complexity: Separations and trade-offs via substitutions. In *Proceedings of the 2nd Symposium on Innovations in Computer Science (ICS '11)*, pages 401–416, January 2011.

11 Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow—resolution made simple. *Journal of the ACM*, 48(2):149–169, March 2001. Preliminary version in *STOC '99*.

12 Ilario Bonacina. Total space in resolution is at least width squared. In *Proceedings of the 43rd International Colloquium on Automata, Languages and Programming (ICALP '16)*, volume 55 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 56:1–56:13, July 2016.

13 Ilario Bonacina and Nicola Galesi. A framework for space complexity in algebraic proof systems. *Journal of the ACM*, 62(3):23:1–23:20, June 2015. Preliminary version in *ITCS '13*.

14 Ilario Bonacina, Nicola Galesi, and Neil Thapen. Total space in resolution. In *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science (FOCS '14)*, pages 641–650, October 2014.

15 María Luisa Bonet, Juan Luis Esteban, Nicola Galesi, and Jan Johannsen. On the relative complexity of resolution refinements and cutting planes proof systems. *SIAM Journal on Computing*, 30(5):1462–1484, 2000. Preliminary version in *FOCS '98*.

16 Siu Man Chan and Aaron Potechin. Tight bounds for monotone switching networks via Fourier analysis. *Theory of Computing*, 10:389–419, October 2014. Preliminary version in *STOC '12*.

17 Ashok K. Chandra. Efficient compilation of linear recursive programs. In *Proceedings of the 14th Annual Symposium on Switching and Automata Theory (SWAT '73)*, pages 16–25, 1973.

18 Vašek Chvátal and Endre Szemerédi. Many hard examples for resolution. *Journal of the ACM*, 35(4):759–768, October 1988.

19 Stephen A. Cook and Robert Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, 44(1):36–50, March 1979.

20 Stephen A. Cook and Ravi Sethi. Storage requirements for deterministic polynomial time recognizable languages. *Journal of Computer and System Sciences*, 13(1):25–37, 1976. Preliminary version in *STOC '74*.

21 Susanna F. de Rezende, Jakob Nordström, and Marc Vinyals. How limited interaction hinders real communication (and what it means for proof and circuit complexity). In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS '16)*, pages 295–304, October 2016.

22 Cynthia Dwork, Moni Naor, and Hoeteck Wee. Pebbling and proofs of work. In *Proceedings of the 25th Annual International Cryptology Conference (CRYPTO '05)*, volume 3621 of *Lecture Notes in Computer Science*, pages 37–54. Springer, August 2005.

23 Patrick W. Dymond and Martin Tompa. Speedups of deterministic machines by synchronous parallel machines. *Journal of Computer and System Sciences*, 30(2):149–161, April 1985. Preliminary version in *STOC '83*.

24 Paul Erdős, Ronald L. Graham, and Endre Szemerédi. On sparse graphs with dense long paths. Technical report, Stanford University, 1975.

25 Juan Luis Esteban and Jacobo Torán. Space bounds for resolution. *Information and Computation*, 171(1):84–97, 2001. Preliminary versions of these results appeared in *STACS '99* and *CSL '99*.

26 Yuval Filmus, Massimo Lauria, Mladen Mikša, Jakob Nordström, and Marc Vinyals. Towards an understanding of polynomial calculus: New separations and lower bounds (Extended abstract). In *Proceedings of the 40th International Colloquium on Automata, Languages and Programming (ICALP '13)*, volume 7965 of *Lecture Notes in Computer Science*, pages 437–448. Springer, July 2013.

27 Yuval Filmus, Massimo Lauria, Jakob Nordström, Noga Ron-Zewi, and Neil Thapen. Space complexity in polynomial calculus. *SIAM Journal on Computing*, 44(4):1119–1153, August 2015. Preliminary version in *CCC '12*.

28 Yuval Filmus, Toniann Pitassi, Robert Robere, and Stephen A Cook. Average case lower bounds for monotone switching networks. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS '13)*, pages 598–607, November 2013.

29 Nicola Galesi, Pavel Pudlák, and Neil Thapen. The space complexity of cutting planes refutations. In *Proceedings of the 30th Annual Computational Complexity Conference (CCC '15)*, volume 33 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 433–447, June 2015.

30 John R. Gilbert and Robert Endre Tarjan. Variations of a pebble game on graphs. Technical Report STAN-CS-78-661, Stanford University, 1978. URL: `http://infolab.stanford.edu/TR/CS-TR-78-661.html`.

31 Mika Göös and Toniann Pitassi. Communication lower bounds via critical block sensitivity. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC '14)*, pages 847–856, May 2014.

32 Armin Haken. The intractability of resolution. *Theoretical Computer Science*, 39(2-3):297–308, August 1985.

33 John Hopcroft, Wolfgang Paul, and Leslie Valiant. On time versus space. *Journal of the ACM*, 24(2):332–337, April 1977. Preliminary version in *FOCS '75*.

34 Trinh Huynh and Jakob Nordström. On the virtue of succinct proofs: Amplifying communication complexity hardness to time-space trade-offs in proof complexity (Extended abstract). In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC '12)*, pages 233–248, May 2012.

35 Matti Järvisalo, Arie Matsliah, Jakob Nordström, and Stanislav Živný. Relating proof complexity measures and practical hardness of SAT. In *Proceedings of the 18th International Conference on Principles and Practice of Constraint Programming (CP '12)*, volume 7514 of *Lecture Notes in Computer Science*, pages 316–331. Springer, October 2012.

36 George Katsirelos, Ashish Sabharwal, Horst Samulowitz, and Laurent Simon. Resolution and parallelizability: Barriers to the efficient parallelization of SAT solvers. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence (AAAI '13)*, July 2013.

37 Thomas Lengauer and Robert Endre Tarjan. Asymptotically tight bounds on time-space trade-offs in a pebble game. *Journal of the ACM*, 29(4):1087–1130, October 1982. Preliminary version in *STOC '79*.

38 João P. Marques-Silva and Karem A. Sakallah. GRASP: A search algorithm for propositional satisfiability. *IEEE Transactions on Computers*, 48(5):506–521, May 1999. Preliminary version in *ICCAD '96*.

39 Jakob Nordström. Narrow proofs may be spacious: Separating space and width in resolution. *SIAM Journal on Computing*, 39(1):59–121, May 2009. Preliminary version in *STOC '06*.

40 Jakob Nordström. On the relative strength of pebbling and resolution. *ACM Transactions on Computational Logic*, 13(2):16:1–16:43, April 2012. Preliminary version in *CCC '10*.

41 Jakob Nordström. Pebble games, proof complexity and time-space trade-offs. *Logical Methods in Computer Science*, 9:15:1–15:63, September 2013.

42 Jakob Nordström. New wine into old wineskins: A survey of some pebbling classics with supplemental results. Manuscript in preparation. To appear in *Foundations and Trends in Theoretical Computer Science*. Current draft version available at `http://www.csc.kth.se/~jakobn/research/`, 2017.

43 Jakob Nordström and Johan Håstad. Towards an optimal separation of space and length in resolution. *Theory of Computing*, 9:471–557, May 2013. Preliminary version in *STOC '08*.

44 Michael S. Paterson and Carl E. Hewitt. Comparative schematology. In *Record of the Project MAC Conference on Concurrent Systems and Parallel Computation*, pages 119–127, 1970.

45 Wolfgang J. Paul and Rüdiger Reischuk. On alternation II. A graph theoretic approach to determinism versus non-determinism. *Acta Informatica*, 14(4):391–403, 1980. Preliminary version in *GITCS '79*.

46 Nicholas Pippenger. Pebbling. Technical Report RC8258, IBM Watson Research Center, 1980. in Proceedings of the 5th IBM Symposium on Mathematical Foundations of Computer Science, Japan.

47 Ran Raz and Pierre McKenzie. Separation of the monotone NC hierarchy. *Combinatorica*, 19(3):403–435, March 1999. Preliminary version in *FOCS '97*.

48 John E. Savage. *Models of Computation: Exploring the Power of Computing*. Addison-Wesley, 1998. URL: `http://www.modelsofcomputation.org`.

49 Georg Schnitger. On depth-reduction and grates. In *Proceedings of the 24th Annual IEEE Symposium on Foundations of Computer Science (FOCS '83)*, pages 323–328, November 1983.

50 Ravi Sethi. Complete register allocation problems. *SIAM Journal on Computing*, 4(3):226–248, September 1975.

51 Sowmitri Swamy and John E. Savage. Space-time trade-offs on the FFT-algorithm. Technical Report CS-31, Brown University, 1977.

52 Alasdair Urquhart. Hard examples for resolution. *Journal of the ACM*, 34(1):209–219, January 1987.

53 Leslie G. Valiant. Graph-theoretic arguments in low-level complexity. In *Proceedings of the 6th International Symposium on Mathematical Foundations of Computer Science (MFCS '77)*, pages 162–176, September 1977.