

Unbounded Discrepancy of Deterministic Random Walks on Grids

Tobias Friedrich^{1(✉)}, Maximilian Katzmann², and Anton Krohmer^{1,2}

¹ Hasso Plattner Institute, Potsdam, Germany
{tobias.friedrich, anton.krohmer}@hpi.de

² Friedrich-Schiller-Universität Jena, Jena, Germany

Abstract. Random walks are frequently used in randomized algorithms. We study a derandomized variant of a random walk on graphs, called rotor-router model. In this model, instead of distributing tokens randomly, each vertex serves its neighbors in a fixed deterministic order. For most setups, both processes behave remarkably similar: Starting with the same initial configuration, the number of tokens in the rotor-router model deviates only slightly from the expected number of tokens on the corresponding vertex in the random walk model. The maximal difference over all vertices and all times is called single vertex discrepancy. Cooper and Spencer (2006) showed that on \mathbb{Z}^d the single vertex discrepancy is only a constant c_d . Other authors also determined the precise value of c_d for $d = 1, 2$. All these results, however, assume that initially all tokens are only placed on one partition of the bipartite graph \mathbb{Z}^d . We show that this assumption is crucial by proving that otherwise the single vertex discrepancy can become arbitrarily large. For all dimensions $d \geq 1$ and arbitrary discrepancies $\ell \geq 0$, we construct configurations that reach a discrepancy of at least ℓ .

1 Introduction

Algorithms that are allowed to make random decision can solve many problems more efficiently than purely deterministic algorithms. One such example is the approximation of the volume of a convex body, where randomness gives a super-polynomial speed-up in computing power [11]. The first polynomial-time algorithm for this (and a number of other) problems is based on a certain random walk (e.g. [1]). Random walks appear to be powerful tools for designing efficient randomized algorithms.

Rotor-Router Model. The wide applicability of random walks raises the question what properties of the random walk are crucial and how much randomness is needed for this. To study this, we consider a derandomized variant of the random walk on the infinite grid \mathbb{Z}^d . In this *rotor-router model*, each vertex $x \in \mathbb{Z}^d$ is equipped with a “rotor” together with a cyclic permutation (called a “rotor sequence”) of the $2d$ cardinal directions of \mathbb{Z}^d . While the tokens performing a random walk leave a vertex in a random direction, in the rotor-router model the tokens deterministically go in the direction the rotor is pointing. After a token

is sent, the rotor is rotated according to the fixed rotor sequence. This ensures that the tokens are distributed evenly among the neighbors.

Synonyms of the Rotor-Router Model. The rotor-router model was rediscovered independently several times in the literature. First under the name “Eulerian walker” [20], then as “edge ant walk” [22] and “whirling tour” [10]. It was later popularized by James Propp [16] and therefore also called “Propp machine” by Cooper and Spencer [6]. The same authors later also used the term “deterministic random walk” [4, 8]. To emphasize the working principle, we only use the term “rotor-router model” in the rest of the paper.

Some Properties of the Rotor-Router Model. Many aspects of the model have been studied. The vertex and edge cover time of the rotor-router model can be asymptotically faster or slower as the classical random walk, depending on the topology [2, 12, 23]. Very precise bounds are also known if multiple tokens are deployed in parallel [7, 15, 17]. Our focus is on the *single-vertex discrepancy* with which we compare the rotor-router model and the expected behavior of the classical random walk. If particles are arbitrarily placed on the vertices and do a simultaneous walk in both models, we are interested in the maximal difference in the number of tokens between both models, at all times and on each vertex.

Known Results for the Single-Vertex Discrepancy. [6] proved that on \mathbb{Z}^d the single vertex discrepancy is a constant c_d . For the case $d = 1$, that is, the graph being the infinite path, Cooper et al. [4] showed that $c_1 \approx 2.29$. For $d = 2$ the constant is $c_2 \approx 7.83$ for circular rotor sequences and $c_2 \approx 7.29$ otherwise [8]. It is further known that there is no such constant for infinite trees [5]. There are also (linear) upper and lower bounds for the discrepancy of finite graphs [14]. For some special finite graphs like hypercubes, stronger (i.e. polylogarithmic in the number of nodes) upper bounds are known [14].

Open Question. All three aforementioned results for the grid \mathbb{Z}^d assume that the initial configuration is “even”, that is, it only has tokens on one partition of the bipartite graph \mathbb{Z}^d . This assumption is, however, essential for achieving a constant discrepancy. Cooper et al. already pointed out for $d = 1$ that without this assumption their results “cannot be expected” [4, p. 2074]. We make this statement rigorous and present for each dimension d a configuration such that the single-vertex discrepancy on \mathbb{Z}^d becomes arbitrarily large.

Results. To allow a direct comparison, let us first restate the result of Cooper and Spencer [6]. The mathematical notation is introduced in Sect. 2.

Theorem 1 ([6]). *For all $d \geq 1$ there is a constant $c_d \in \mathbb{R}_+$ such that for all even initial configurations, the single-vertex discrepancy on \mathbb{Z}^d is bounded by c_d .*

Our main result is the following complement of the previous statement.

Theorem 2. *For all $d \geq 1$ and $\ell \in \mathbb{R}$ there is an initial configuration such that the single-vertex discrepancy on \mathbb{Z}^d is at least ℓ .*

The reason for the unbounded discrepancy observed for non-even initial configurations is that the two partitions of \mathbb{Z}^d subtly interfere with each other

through the rotors. In every time step, all tokens switch back and forth between even and odd positions. In a random walk they are distributed independently, in the rotor-router model they follow the rotors, which exchange information between both partitions. This causes the unbounded discrepancy for appropriately set up initial configurations.

It should be noted that the discrepancy of ℓ in Theorem 2 already occurs for small configurations. In fact, Corollary 8 shows that a discrepancy of ℓ can be reached after $\Theta(\lceil \ell^2/d^2 \rceil)$ time steps with $\mathcal{O}(\lceil 1 + \ell/d \rceil^{2d+1})$ tokens.

Techniques. For proving Theorem 2, we define a specific (infinitely large) initial configuration called (k, d) -wedge (cf. Definition 4), for which we study explicitly how it develops over time in the rotor-router and random walk model. We prove that this configuration is “stable” in the rotor-router model, that is, it stays unchanged after an even number of steps (cf. Lemma 6). The proof needs to consider 26 cases. We prove the cases using an automated theorem prover. Given this structural insight on the behavior of (k, d) -wedge, we calculate the resulting discrepancy (cf. Lemma 7). The proof makes use of the fact that the expected behavior of the d -dimensional random walk starting with a (k, d) -wedge can be decomposed into a collection of 1-dimensional random walks. To obtain a result for finite time and finite configurations, we observe that a subset of the (k, d) -wedge suffices to achieve a desired discrepancy (cf. Corollary 8).

2 Preliminaries

Random Walks. A random walk is a stochastic process that describes the movement of a number of tokens on a graph G . At each time step, each token at a vertex \mathbf{x} chooses a neighbor independently and uniformly at random, and moves to that neighbor.

We consider simple random walks on an infinite d -dimensional grid \mathbb{Z}^d . A token at coordinate $\mathbf{x} = (x_1, \dots, x_d)$ can move in the $2d$ cardinal directions, as given by the unit vectors: $\mathbf{e}_1 = (1, 0, 0, \dots), \mathbf{e}_2 = (0, 1, 0, \dots), \dots, -\mathbf{e}_1 = (-1, 0, 0, \dots), -\mathbf{e}_2 = (0, -1, 0, \dots), \dots, -\mathbf{e}_d = (0, \dots, -1)$. We refer to this set of directions by E_{2d} . Following [18], we write Z_i for the direction that a token took at time step i . As all directions are equiprobable and independent, we have $\Pr[Z_i = \mathbf{e}_j] = \Pr[Z_i = -\mathbf{e}_j] = \frac{1}{2d}$ for all j . The position of a token after t steps can then be described as a sum of random variables $S_t = \mathbf{x} + Z_1 + Z_2 + \dots + Z_t$.

We write $S_t^d(\mathbf{x})$ to express the probability that a d -dimensional random walk starting at the origin reaches vertex \mathbf{x} after t steps. E.g., for dimension $d = 1$ we obtain $S_t^1(x) = 2^{-t} \binom{t}{(t+x)/2}$.

We denote by $\bar{\mathbf{x}}$ the sum of the individual components of \mathbf{x} , i.e. $\bar{\mathbf{x}} := \mathbf{x}^T \mathbf{1} = \sum_{i=1}^d x_i$. Observe that the grid \mathbb{Z}^d is a bipartite graph where all nodes with even $\bar{\mathbf{x}}$ form one partition, and nodes with odd $\bar{\mathbf{x}}$ form the other. With each time step, a token therefore switches the partition. To this end, we have $S_t^d(\mathbf{x}) = 0$ if $(\bar{\mathbf{x}} - t \equiv 1) \pmod 2$. We write $a \sim t$ to say that $(a \equiv t) \pmod 2$, and we call a node \mathbf{x} *even* if $\bar{\mathbf{x}} \sim 0$, and *odd* otherwise.

Rotor-Router Model. Let us now formally define the rotor-router model on the grid \mathbb{Z}^d . Each vertex \mathbf{x} in this graph is equipped with a *rotor* $r_{\mathbf{x}} \in E_{2d}$. The *rotor sequence* for a vertex \mathbf{x} is defined by a cyclic permutation $r_{\mathbf{x}}^{\rightarrow}: E_{2d} \rightarrow E_{2d}$.

At each time step t , all tokens at \mathbf{x} do exactly one move as follows. A particular token moves in the direction of the rotor $r_{\mathbf{x}}$; and afterwards, the rotor is updated to point to $r_{\mathbf{x}}^{\rightarrow}(r_{\mathbf{x}})$. This is repeated until all tokens have been moved. Since tokens are not labeled, the order in which the tokens are passed to the rotor does not matter. All configurations of the rotor-router model are therefore fully defined by the initial placement of tokens, the initial rotor configurations $r_{\mathbf{x}}$ and the rotor sequences $r_{\mathbf{x}}^{\rightarrow}$ for all vertices $\mathbf{x} \in \mathbb{Z}^d$. If all tokens are initially on even vertices, we speak of an *even configuration*.

Single Vertex Discrepancy. When comparing the quality of the simulation of the rotor-router model, one often refers to the *single vertex discrepancy*, which is defined as follows. Let $f(\mathbf{x}, t): \mathbb{Z}^d \times \mathbb{N}_0 \rightarrow \mathbb{N}_0$ be the number of tokens at vertex \mathbf{x} after t steps of the (deterministic) rotor-router model, and let $\mathbb{E}(\mathbf{x}, t): \mathbb{Z}^d \times \mathbb{N}_0 \rightarrow \mathbb{R}^+$ denote the expected number of tokens after t steps of a random walk with the same starting configuration $f(\mathbf{x}, 0)$. To compute $\mathbb{E}(\mathbf{x}, t)$ we determine for each $\mathbf{y} \in \mathbb{Z}^d$ the probability that a random walk starting at \mathbf{y} reaches \mathbf{x} after exactly t steps and multiply the result with the number of tokens that were at \mathbf{y} . Hence,

$$\mathbb{E}(\mathbf{x}, t) = \sum_{\mathbf{y} \in \mathbb{Z}^d} f(\mathbf{y}, 0) \cdot S_t^d(\mathbf{x} - \mathbf{y}). \quad (1)$$

Using this, we can define the *single vertex discrepancy*.

Definition 3. Let $d \geq 1$ and an initial configuration $f(\mathbf{x}, 0)$ for all $\mathbf{x} \in \mathbb{Z}^d$ be given. We call $\Delta(\mathbf{x}, t) = |f(\mathbf{x}, t) - \mathbb{E}(\mathbf{x}, t)|$ the single vertex discrepancy at \mathbf{x} after t steps. Then, we define the single vertex discrepancy Δ_d as

$$\Delta_d := \sup_{\mathbf{x} \in \mathbb{Z}^d, t \in \mathbb{N}} \Delta(\mathbf{x}, t). \quad (2)$$

3 Stable Configuration of the Rotor-Router Model

According to Theorem 1, the single vertex discrepancy is constant if we start with an even configuration. To prove that this condition is necessary, we construct the (k, d) -wedge, a starting configuration of tokens that ensures that there are effectively only two states of the rotor-router model.

The (k, d) -wedge intuitively forms a “peak” of tokens at the origin, and the rest of the graph is populated with tokens in a way that stabilizes the peak. In the random walk model, the expected number of nodes in the origin will decrease over time, while in the rotor-router model, the number of nodes always stays the same. The (k, d) -wedge is illustrated in Fig. 1 and formally defined as follows.

Definition 4. Let $k, d \in \mathbb{N}$ be given, where k adjusts the vertex discrepancy. The rotor direction of vertex \mathbf{x} at time t will be referred to by $r(\mathbf{x}, t): \mathbb{Z}^d \times \mathbb{N}_0 \rightarrow E_{2d}$.

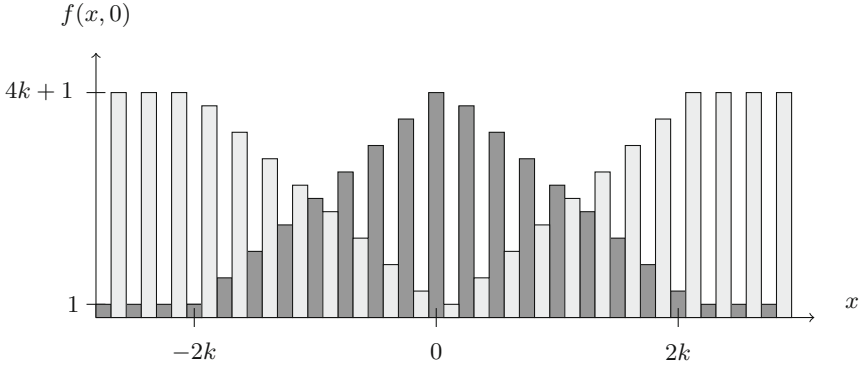


Fig. 1. Illustration of the $(k, 1)$ -wedge in dimension 1. The y -axis describes the number of tokens at position x . Dark colored bars show the even partition, light colored bars the odd one. This stable configuration is used to show our main result.

We define the (k, d) -wedge, a starting configuration of the rotor-router model, as follows. For even vertices \mathbf{x} with $\bar{\mathbf{x}} \sim 0$, we set

$$f(\mathbf{x}, 0) := f_0(\bar{\mathbf{x}}, 0) := \begin{cases} d \cdot (4k + 1 + 2\bar{\mathbf{x}}) & \text{if } \bar{\mathbf{x}} \in [-2k, 0], \\ d \cdot (4k + 3 - 2\bar{\mathbf{x}}) & \text{if } \bar{\mathbf{x}} \in [1, 2k], \\ d & \text{otherwise.} \end{cases}$$

$$r(\mathbf{x}, 0) := r_0(\bar{\mathbf{x}}, 0) := \begin{cases} -\mathbf{e}_1 & \text{if } \bar{\mathbf{x}} \in [1, 2k], \\ \mathbf{e}_1 & \text{otherwise.} \end{cases}$$

For odd vertices \mathbf{x} with $\bar{\mathbf{x}} \sim 1$, we set

$$f(\mathbf{x}, 0) := f_1(\bar{\mathbf{x}}, 0) := \begin{cases} d \cdot (1 - 2\bar{\mathbf{x}}) & \text{if } \bar{\mathbf{x}} \in [-2k, 0], \\ d \cdot (2\bar{\mathbf{x}} - 1) & \text{if } \bar{\mathbf{x}} \in [1, 2k], \\ d \cdot (4k + 1) & \text{otherwise.} \end{cases}$$

$$r(\mathbf{x}, 0) := r_1(\bar{\mathbf{x}}, 0) := \begin{cases} -\mathbf{e}_1 & \text{if } \bar{\mathbf{x}} \in [-2k, -1], \\ \mathbf{e}_1 & \text{otherwise.} \end{cases}$$

The rotor sequences follow the order $\mathbf{e}_1, \dots, \mathbf{e}_d, -\mathbf{e}_1, \dots, -\mathbf{e}_d$.

Next, we show that the (k, d) -wedge is a stable configuration, meaning that the rotor-router model returns to the initial configuration every two steps. To this end, we introduce a function $g: \mathbb{Z}^d \times E_{2d} \times E_{2d} \times \mathbb{N} \rightarrow \mathbb{N}$, where $g(\mathbf{x}, \pm \mathbf{e}_i, \pm \mathbf{e}_j, t)$ denotes the number of tokens that vertex \mathbf{x} receives from vertex $\mathbf{x} \pm \mathbf{e}_i$ at time t when $r(\mathbf{x} \pm \mathbf{e}_i, t) = \pm \mathbf{e}_j$. Therefore,

$$g(\mathbf{x}, \mathbf{e}, \mathbf{f}, t) = \begin{cases} \frac{f(\mathbf{x} + \mathbf{e}, t) - d}{2d} & \text{if } \text{sgn}(\mathbf{e}) = \text{sgn}(\mathbf{f}), \\ \frac{f(\mathbf{x} + \mathbf{e}, t) + d}{2d} & \text{otherwise,} \end{cases} \tag{3}$$

where $\text{sgn}(-\mathbf{e}_i) = -1$ and $\text{sgn}(\mathbf{e}_i) = 1$ for all $i = 1, \dots, d$. Then we can write

$$f(\mathbf{x}, t+1) = \sum_{i=1}^d g(\mathbf{x}, \mathbf{e}_i, r(\mathbf{x} + \mathbf{e}_i, t), t) + \sum_{i=1}^d g(\mathbf{x}, -\mathbf{e}_i, r(\mathbf{x} - \mathbf{e}_i, t), t), \quad (4)$$

which results from summing up the number of tokens that the neighbors of \mathbf{x} pass to \mathbf{x} at time step t . Recall that $f(\mathbf{x}, 0) = f(\bar{\mathbf{x}}, 0)$ and therefore $f(\mathbf{x} \pm \mathbf{e}_1, 0) = f(\bar{\mathbf{x}} \pm 1, 0)$. The same holds for $r(\mathbf{x}, 0)$. The definition of g in Eq. (3) can in this case be extended to $g(\bar{\mathbf{x}}, \pm 1, \pm \mathbf{e}_1, 0)$, and we can simplify Eq. (4) to

$$\begin{aligned} f(\bar{\mathbf{x}}, 1) &= \sum_{i=1}^d g(\bar{\mathbf{x}}, 1, r(\bar{\mathbf{x}} + 1, 0), 0) + \sum_{i=1}^d g(\bar{\mathbf{x}}, -1, r(\bar{\mathbf{x}} - 1, 0), 0) \\ &= d \cdot (g(\bar{\mathbf{x}}, 1, r(\bar{\mathbf{x}} + 1, 0), 0) + g(\bar{\mathbf{x}}, -1, r(\bar{\mathbf{x}} - 1, 0), 0)). \end{aligned} \quad (5)$$

To prove stability, it remains to show the following Lemmata.

Lemma 5. *Given a (k, d) -wedge, it holds*

$$r(\mathbf{x}, 1) = -r(\mathbf{x}, 0) \quad \text{and} \quad f(\mathbf{x}, 1) = \begin{cases} f_1(\bar{\mathbf{x}}, 0) & \text{if } \bar{\mathbf{x}} \sim 0, \\ f_0(\bar{\mathbf{x}}, 0) & \text{if } \bar{\mathbf{x}} \sim 1. \end{cases}$$

Lemma 6. *Given a (k, d) -wedge, it holds $r(\mathbf{x}, 2) = r(\mathbf{x}, 0)$ and $f(\mathbf{x}, 2) = f(\mathbf{x}, 0)$.*

Lemma 5 states that the configuration of the rotor-router model after one step is again the (k, d) -wedge, except that it is shifted by one to the left. Furthermore, all rotors point in the opposite direction. By the same intuition, the next step undoes these changes and the configuration returns to the (k, d) -wedge after 2 steps, which is shown by Lemma 6.

These statements can be proven by a case distinction over Eq. (5). While none of the cases are mathematically challenging, there are 26 of them. Proving every case by hand is tedious and provides little to no further insight to the problem. Nevertheless, even small off-by-one errors break the stability of the (k, d) -wedge, which is why we wanted to convince ourselves that the (k, d) -wedge is indeed correct. To this end, we used the automated prover Isabelle/HOL [19] for the case distinction. Our code can be found in the long version of this paper.

Such provers excel at keeping track of all subgoals (i.e. cases) of a proof. Mostly, the proofs are not human readable, as they rely on internal proof routines. Automated proof systems like Isabelle/HOL, however, contain a certified kernel; so trusting the automated proof boils down to trusting the formalization of the problem and the correctness of the kernel. It is debated whether an automated proof can be considered correct or not—in our case, we believe that it is more reasonable to trust the correctness of Isabelle’s kernel than to trust a lengthy and error-prone proof of 26 cases.

Discrepancy with Infinite Steps. If the rotor-router model is initialized with the (k, d) -wedge, the number of tokens stays the same at all vertices \mathbf{x} , independent of the number of steps the process is run (mod 2), as was shown above. In contrast, the expected number of tokens on the even partition decreases over time for the random walk. The reason for this is that at every time step and on every vertex the number of tokens is not a multiple of the number of neighboring vertices, ensuring that the rotor-router model cannot distribute the tokens equally to all neighbors as the random walk does. To show a lower bound on the discrepancy, we inspect the difference between the actual and the expected number of tokens at the origin after enough steps. We prove the following lemma.

Lemma 7. *If the rotor-router model is initialized with the (k, d) -wedge, we have*

$$\lim_{t \rightarrow \infty} \Delta(0, t) \geq 4dk.$$

Proof. Recall that $f(0, t)$ describes the number of tokens at $\mathbf{x} = 0$ when the rotor-router model is run, whereas $\mathbb{E}(0, t)$ describes the expected number of tokens at $\mathbf{x} = 0$ for the random walk after t steps. By Definition 3,

$$\Delta(0, t) = |f(0, t) - \mathbb{E}(0, t)|.$$

For the sake of brevity, we assume from now on that t is even; however, the statement holds for all t . Then, since the (k, d) -wedge was proven to be stable, we obtain $f(0, t) = d \cdot (4k + 1)$.

The calculation of $\mathbb{E}(0, t)$ is more involved. According to Eq. (1),

$$\mathbb{E}(0, t) = \sum_{\mathbf{y} \in \mathbb{Z}^d} f(\mathbf{y}, 0) \cdot S_t^d(\mathbf{y}),$$

where $S_t^d(\mathbf{y})$ is the probability that a d -dimensional random walk that starts at $\mathbf{y} = (y_1, \dots, y_d)$ ends at 0 after t steps. $S_t^d(\mathbf{y})$ admits simple formulas for $d \in \{1, 2\}$, but there are no simple equations for $d \geq 3$ known to us.

To circumvent this problem, we show that the expected number of tokens $\mathbb{E}(\mathbf{x}, t)$ is actually the same for all dimensions $d \geq 1$; if the starting configuration is the (k, d) -wedge.

Consider the expected number of tokens at a vertex \mathbf{x} with respect to $\bar{\mathbf{x}} = x_1 + \dots + x_d$. With one step, a token starting at \mathbf{x} can only reach vertices \mathbf{y} with $\bar{\mathbf{y}} \in \{\bar{\mathbf{x}} - 1, \bar{\mathbf{x}} + 1\}$. The probability that either happens is $1/2$, i.e.

$$\sum_{\substack{\mathbf{y} \in \mathbb{Z}^d \\ \bar{\mathbf{y}} = b}} S_1^d(\mathbf{x} - \mathbf{y}) = \begin{cases} \frac{1}{2}, & \text{if } b \in \{\bar{\mathbf{x}} - 1, \bar{\mathbf{x}} + 1\} \\ 0 & \text{otherwise.} \end{cases}$$

Consider now the following variation of a random walk on \mathbb{Z}^d , where each token can only move in one dimension, i.e.

$$\begin{aligned} \Pr[Z_i = \mathbf{e}_1] &= \Pr[Z_i = -\mathbf{e}_1] = 1/2, \\ \Pr[Z_i = \mathbf{e}_j] &= \Pr[Z_i = -\mathbf{e}_j] = 0 \quad \text{for all } j > 1. \end{aligned}$$

In this setting, we obtain a collection of 1-dimensional random walks operating independently of each other. We write $\mathbb{E}'(\mathbf{x}, t)$ to denote the expected number of tokens in this random walk; and we initialize $\mathbb{E}'(\mathbf{x}, 0)$ again with the (k, d) -wedge. Note that $\mathbb{E}'(\mathbf{x}, t) = \mathbb{E}'(\bar{\mathbf{x}}, t)$ again only depends on $\bar{\mathbf{x}}$ and t . By showing $\mathbb{E}'(\mathbf{x}, t) = \mathbb{E}(\mathbf{x}, t)$ we can analyze a 1-dimensional random walk and directly obtain results for d -dimensional random walks.

We prove $\mathbb{E}'(\mathbf{x}, t) = \mathbb{E}(\mathbf{x}, t)$ by induction over t . For the base case, we have $\mathbb{E}(\mathbf{x}, 0) = \mathbb{E}'(\mathbf{x}, 0)$ by definition. For the inductive step $t \rightarrow t + 1$, we obtain

$$\begin{aligned} \mathbb{E}(\mathbf{x}, t) &= \sum_{\mathbf{y} \in \mathbb{Z}^d} \mathbb{E}(\mathbf{y}, t - 1) \cdot S_1^d(\mathbf{x} - \mathbf{y}) & (6) \\ &= \sum_{\substack{\mathbf{y} \in \mathbb{Z}^d \\ \bar{\mathbf{y}} = \bar{\mathbf{x}} + 1}} \mathbb{E}'(\bar{\mathbf{y}}, t - 1) \cdot S_1^d(\mathbf{x} - \mathbf{y}) + \sum_{\substack{\mathbf{y} \in \mathbb{Z}^d \\ \bar{\mathbf{y}} = \bar{\mathbf{x}} - 1}} \mathbb{E}'(\bar{\mathbf{y}}, t - 1) \cdot S_1^d(\mathbf{x} - \mathbf{y}) \\ &= \mathbb{E}'(\bar{\mathbf{x}} + 1, t - 1) \cdot \frac{1}{2} + \mathbb{E}'(\bar{\mathbf{x}} - 1, t - 1) \cdot \frac{1}{2} \\ &= \mathbb{E}'(\bar{\mathbf{x}}, t) = \mathbb{E}'(\mathbf{x}, t), & (7) \end{aligned}$$

where Eqs. (6) and (7) hold by the tower rule for expectation.

We now focus on the 1-dimensional random walk initialized with the (k, d) -wedge. Let $I_1 := [-2k, 2k]$ and $I_2 := \mathbb{Z} \setminus I_1$. We know that $f(\mathbf{x}, t) = d$ for all $x \in I_2, x \sim 0$. We denote the expected number of tokens that started in $S \subseteq \mathbb{Z}$ and arrive at the origin after $t \sim 0$ steps by $\mathbb{E}_S(0, t)$.

$$\mathbb{E}_{I_2}(0, t) = \sum_{\substack{x \in I_2 \\ x \sim 0}} f(x, 0) \cdot S_t^1(|x|) \leq \sum_{\substack{x \in [-t, t] \\ x \sim 0}} d \cdot 2^{-t} \cdot \binom{t}{(t + |x|)/2}.$$

We now split the sum using that $S_t^1(x) = S_t^1(-x)$:

$$\mathbb{E}_{I_2}(0, t) \leq \frac{d}{2^t} \cdot \left(\sum_{\substack{x=0 \\ x \sim 0}}^t \binom{t}{(t+x)/2} + \sum_{\substack{x=2 \\ x \sim 0}}^t \binom{t}{(t+x)/2} \right) = \frac{d}{2^t} \cdot \sum_{x=0}^t \binom{t}{x} = d.$$

This approximation shows that $\mathbb{E}_{I_2}(0, t) \leq d$, which is obviously independent of the number of steps the process is run.

The number of expected tokens that started in I_1 and end at the origin after t steps will be approximated using the upper bound $\binom{t}{t/2} \leq \sqrt{\frac{2}{\pi t}} \cdot 2^t \cdot e^{-\frac{18t-1}{72t^2+12t}}$ [21]. Then, \mathbb{E}_{I_1} can be estimated the following way:

$$\begin{aligned} \mathbb{E}_{I_1}(0, t) &= \sum_{i=1}^k S_t^1(2i) \cdot f(2i, 0) + \sum_{i=0}^k S_t^1(2i) \cdot f(-2i, 0) \\ &= d2^{-t} \left(\sum_{i=1}^k \binom{t}{\frac{t}{2} + i} \cdot (4k + 3 - 4i) + \sum_{i=0}^k \binom{t}{\frac{t}{2} + i} \cdot (4k + 1 - 4i) \right) \end{aligned}$$

$$\begin{aligned} &\leq \binom{t}{t/2} \cdot d2^{-t} \cdot \left(\sum_{i=1}^k (4k + 3 - 4i) + \sum_{i=0}^k (4k + 1 - 4i) \right) \\ &= \binom{t}{t/2} \cdot d2^{-t} \cdot (2k + 1)^2 \leq \sqrt{\frac{2}{\pi t}} \cdot e^{-\frac{18t-1}{72t^2+12t}} \cdot d \cdot (2k + 1)^2. \end{aligned}$$

Knowing $\mathbb{E}_{I_1}(0, t)$ and $\mathbb{E}_{I_2}(0, t)$, we compute $\mathbb{E}(0, t)$ by adding these terms and obtain $\mathbb{E}(0, t) \leq d + \sqrt{\frac{2}{\pi t}} \cdot e^{-\frac{18t-1}{72t^2+12t}} \cdot d \cdot (2k + 1)^2$. This results in a discrepancy of

$$|f(0, t) - E(0, t)| \geq \max \left\{ 0, 4dk - \sqrt{\frac{2}{\pi t}} \cdot e^{-\frac{18t-1}{72t^2+12t}} \cdot d \cdot (2k + 1)^2 \right\}. \quad (8)$$

For large enough t , this proves the claim. □

This means that by using the second partition of \mathbb{Z}^d in the rotor-router model, it is possible to produce an arbitrarily large discrepancy of $\Omega(dk)$ which reveals that there is no constant bound for the single vertex discrepancy. Figure 2 illustrates the single vertex discrepancy in a $(k, 1)$ -wedge over time for $k \in \{16, 32, 64\}$.

Discrepancy Within Finite Steps. Lemma 7 shows that a discrepancy of $4dk$ can be reached if the processes are run for $t \rightarrow \infty$ steps. It is, however, possible to achieve high discrepancy using already few steps by investigating Eq. (8) more carefully. We show the following Corollary.

Corollary 8. *Given dimension $d \geq 1$ and a discrepancy $\ell \in \mathbb{R}_+$, there exists a (k, d) -wedge that reaches the discrepancy ℓ in $t \in \mathcal{O}(\lceil \ell^2/d^2 \rceil)$ steps using $\mathcal{O}(\lceil 1 + \ell/d \rceil^{2d+1})$ tokens.*

Proof. By Eq. (8), the number of steps that are needed to reach discrepancy ℓ with a (k, d) -wedge are

$$\begin{aligned} \ell &\stackrel{!}{\leq} 4dk - \sqrt{\frac{2}{\pi t}} \cdot e^{-\frac{18t-1}{72t^2+12t}} \cdot d \cdot (2k + 1)^2 \\ \Leftrightarrow t &\geq \frac{2}{\pi} \cdot \frac{d^2(2k + 1)^4}{(4dk - \ell)^2} \end{aligned}$$

Using standard analysis tools, we find that the minimum number of steps necessary to reach the given discrepancy ℓ is

$$t = \frac{2 \cdot d^2(\lceil \frac{d+\ell}{2d} \rceil + 1)^4}{\pi \cdot (2d + \ell)^2} \in \Theta \left(\left\lceil \frac{\ell^2}{d^2} \right\rceil \right)$$

when using a $(\lceil \frac{d+\ell}{2d} \rceil, d)$ -wedge. As the process runs t steps, it visits $\Theta(t^d)$ positions of the grid \mathbb{Z}^d , each of which needs $\leq d \cdot (4k + 1)$ tokens. Therefore, in total it needs at most $\mathcal{O}(\lceil 1 + \ell/d \rceil^{2d+1})$ tokens. □

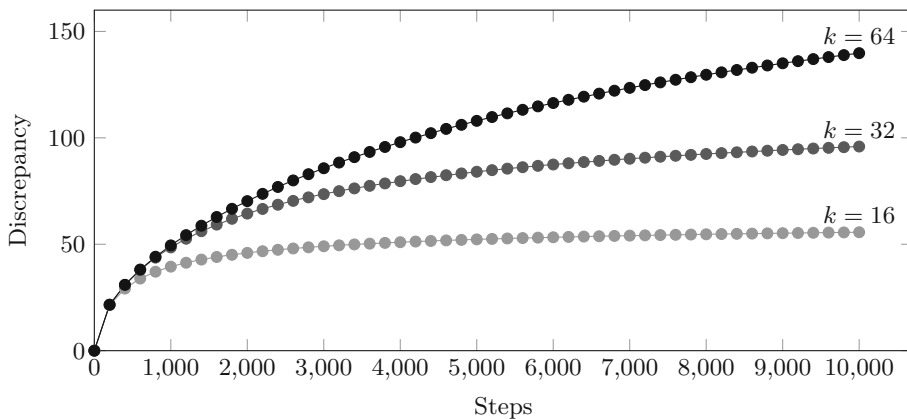


Fig. 2. The simulated single vertex discrepancies for different $(k, 1)$ -wedges. The plots show that even for small t and k a high discrepancy can be achieved. This intuition is formalized in Corollary 8.

4 Conclusion

The rotor-router model is a derandomized variant of the classical random walk. It can be used algorithmically for example in broadcasting [9], external mergesort [3] and load balancing [13]. We study the similarity of the rotor-router model to the expected behavior of the random walk. It was observed and well studied that on grids the number of tokens only differs by some small constant at all times and on each vertex [4, 6, 8]. We closely look at the underlying assumptions of these results and prove that if tokens are allowed to start at an arbitrary position, both models can deviate arbitrarily far. Besides the revealed combinatorial structure, our result indicates that also in algorithmic applications the rotor-router model can deviate significantly from the expected behavior of the random walk, which should be studied further.

References

1. Aleliunas, R., Karp, R.M., Lipton, R.J., Lovasz, L., Rackoff, C.: Random walks, universal traversal sequences, and the complexity of maze problems. In: 20th FOCS, pp. 218–223 (1979)
2. Bampas, E., Gašieniec, L., Hanusse, N., Ilcinkas, D., Klasing, R., Kosowski, A.: Euler tour lock-in problem in the rotor-router model. In: Keidar, I. (ed.) DISC 2009. LNCS, vol. 5805, pp. 423–435. Springer, Heidelberg (2009)
3. Barve, R.D., Grove, E.F., Vitter, J.S.: Simple randomized mergesort on parallel disks. *Parallel Comput.* **23**, 601–631 (1997). Also in 8th SPAA, pp. 109–118 (1996)
4. Cooper, J., Doerr, B., Spencer, J., Tardos, G.: Deterministic random walks on the integers. *Eur. J. Combin.* **28**, 2072–2090 (2007). Also in 3rd ANALCO, pp. 185–197 (2006)

5. Cooper, J., Doerr, B., Friedrich, T., Spencer, J.: Deterministic random walks on regular trees. *Random Struct. Algorithms* **37**, 353–366 (2010). Also in 19th SODA, pp. 766–772 (2008)
6. Cooper, J.N., Spencer, J.: Simulating a random walk with constant error. *Combin. Probab. Comput.* **15**, 815–822 (2006)
7. Dereniowski, D., Kosowski, A., Pajak, D., Uznanski, P.: Bounds on the cover time of parallel rotor walks. In: 31st STACS, pp. 263–275 (2014)
8. Doerr, B., Friedrich, T.: Deterministic random walks on the two-dimensional grid. *Combin. Probab. Comput.* **18**, 123–144 (2009). Also in 17th ISAAC, pp. 474–483 (2006)
9. Doerr, B., Friedrich, T., Sauerwald, T.: Quasirandom rumor spreading. *ACM Trans. Algorithms* **11**, 9:1–9:35 (2014). Also in 19th SODA, pp. 773–781 (2008)
10. Dumitriu, I., Tetali, P., Winkler, P.: On playing golf with two balls. *SIAM J. Discrete Math.* **16**, 604–615 (2003)
11. Dyer, M., Frieze, A., Kannan, R.: A random polynomial-time algorithm for approximating the volume of convex bodies. *J. ACM* **38**, 1–17 (1991). Also in 21st STOC, pp. 375–381 (1989)
12. Friedrich, T., Sauerwald, T.: The cover time of deterministic random walks. In: Thai, M.T., Sahni, S. (eds.) *COCOON 2010*. LNCS, vol. 6196, pp. 130–139. Springer, Heidelberg (2010)
13. Friedrich, T., Gairing, M., Sauerwald, T.: Quasirandom load balancing. *SIAM J. Comput.* **41**, 747–771 (2012). Also in 21st SODA, pp. 1620–1629 (2010)
14. Kijima, S., Koga, K., Makino, K.: Deterministic random walks on finite graphs. In: 9th ANALCO, pp. 16–25 (2012)
15. Klasing, R., Kosowski, A., Pajak, D., Sauerwald, T.: The multi-agent rotor-router on the ring: a deterministic alternative to parallel random walks. In: 32nd PODC, pp. 365–374 (2013)
16. Kleber, M.: Goldbug variations. *The Math. Intell.* **27**, 55–63 (2005)
17. Kosowski, A., Pajak, D.: Does adding more agents make a difference? a case study of cover time for the rotor-router. In: Esparza, J., Fraigniaud, P., Husfeldt, T., Koutsoupias, E. (eds.) *ICALP 2014, Part II*. LNCS, vol. 8573, pp. 544–555. Springer, Heidelberg (2014)
18. Lawler, G., Limic, V.: *Random Walk: A Modern Introduction*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, Cambridge (2010)
19. Nipkow, T., Paulson, L.C., Wenzel, M. (eds.): *Isabelle/HOL: A Proof Assistant for Higher-Order Logic*. LNCS, vol. 2283. Springer, Heidelberg (2002)
20. Priezzhev, V.B., Dhar, D., Dhar, A., Krishnamurthy, S.: Eulerian walkers as a model of self-organized criticality. *Phys. Rev. Lett.* **77**, 5079–5082 (1996)
21. Robbins, H.: A remark on Stirling’s formula. *The Am. Math. Mon.* **62**, 26–29 (1955)
22. Wagner, I.A., Lindenbaum, M., Bruckstein, A.M.: Distributed covering by ant-robots using evaporating traces. *IEEE Trans. Rob. Autom.* **15**, 918–933 (1999)
23. Yanovski, V., Wagner, I.A., Bruckstein, A.M.: A distributed ant algorithm for efficiently patrolling a network. *Algorithmica* **37**, 165–186 (2003)