SIAM J. DISCRETE MATH.
Vol. 32, No. 4, pp. 2441–2452

UNBOUNDED DISCREPANCY OF DETERMINISTIC RANDOM WALKS ON GRIDS*

TOBIAS FRIEDRICH†, MAXIMILIAN KATZMANN†, AND ANTON KROHMER†

Abstract. Random walks are frequently used in randomized algorithms. We study a derandomized variant of a random walk on graphs called the 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 in a remarkably similar way: 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 [*Combin. Probab. Comput.*, 15 (2006), pp. 815–822] 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 of 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 $d \geqslant 1$ and arbitrary discrepancies $\ell \geqslant 0$, we construct configurations that reach a discrepancy of at least ℓ.

Key words. deterministic random walk, rotor-router model, single vertex discrepancy

AMS subject classification. 60G50

DOI. 10.1137/17M1131088

1. Introduction.

Algorithms that are allowed to make random decisions can solve many problems more efficiently than purely deterministic algorithms. One example is the approximation of the volume of a convex body, where randomness gives a superpolynomial speed-up in computing power [11]. The first polynomial-time algorithm for this and 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 of what properties of the random walk are crucial and how much randomness is needed. 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 $\vec{x} \in \mathbb{Z}^d$ 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" [21], then as "edge ant walk" [23], and then "whirling tour" [10]. It was later popularized by James Propp [17] and, therefore, also called "Propp machine" by Cooper and Spencer [6]. The same authors later also used the term "deterministic random walk" [5, 8]. To emphasize the working principle, we only use the term

*Received by the editors May 19, 2017; accepted for publication (in revised form) July 25, 2018; published electronically October 18, 2018. A preliminary version of this paper appeared in [13].
http://www.siam.org/journals/sidma/32-4/M113108.html

†Algorithm Engineering, Hasso Plattner Institute, University of Potsdam, Potsdam 14482, Germany (tobias.friedrich@hpi.de, maximilian.katzmann@hpi.de, anton.krohmer@hpi.de).

"rotor-router model" in the rest of this 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 than the classical random walk, depending on the topology [14, 2, 24]. Very precise bounds are also known if multiple tokens are deployed in parallel [16, 18, 7]. 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. Cooper and Spencer [6] proved that on $\mathbb{Z}^d$, the single vertex discrepancy is a constant c_d. For the case $d = 1$, that is, when the graph is the infinite path, Cooper et al. [5] 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 [4]. There are also (linear) upper and lower bounds for the discrepancy of finite graphs [15]. For some special finite graphs like hypercubes, stronger (i.e., polylogarithmic in the number of nodes) upper bounds are known [15].

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" [5, 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 section 2.

THEOREM 1.1 (see [6]). *For all $d \geqslant 1$ 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 1.2. *For all $d \geqslant 1$ 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 noneven 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 an unbounded discrepancy for appropriately set up initial configurations.

It should be noted that the discrepancy of ℓ in Theorem 1.2 already occurs for small configurations. In fact, Corollary 3.5 shows that a discrepancy of ℓ can be reached after $\Theta\left(\lceil \ell^2/d^2 \rceil\right)$ time steps with $\mathcal{O}(\lceil 1 + \ell/d \rceil^{2d+1})$ tokens.

Techniques. For proving Theorem 1.2, we define a specific (infinitely large) initial configuration called (k, d)-wedge (cf. Definition 3.1), 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 3.3). The proof needs to consider 26 cases. We only present three of them and verify the remaining cases with an automated theorem prover (in the supplementary section SM1 of this paper). Given this structural insight on the behavior of the (k,d)-wedge, we calculate the resulting discrepancy (cf. Lemma 3.4). 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 3.5).

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 $\vec{x}$ 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 $\vec{x} = (x_1, \ldots, x_d)$ can move in one of the $2d$ cardinal directions, as given by the unit vectors: $\vec{e}_1 = (1,0,0,\ldots), \vec{e}_2 = (0,1,0,\ldots), \ldots, -\vec{e}_1 = (-1,0,0,\ldots), -\vec{e}_2 = (0,-1,0,\ldots), \ldots, -\vec{e}_d = (0,\ldots,-1)$. We refer to this set of directions by E_{2d}. Following [19], we write Z_i for the direction that a token took at time step i. As all directions are equiprobable and independent, we have $\Pr[Z_i = \vec{e}_j] = \Pr[Z_i = -\vec{e}_j] = \frac{1}{2d}$ for all j. The position of a token after t steps can then be described as a sum of random variables $S_t = \vec{x} + Z_1 + Z_2 + \cdots + Z_t$.

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

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

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

At each time step t, all tokens at $\vec{x}$ do exactly one move as follows. A particular token moves in the direction of the rotor $r_{\vec{x}}$, and afterwards, the rotor is updated to point to $\rho_{\vec{x}}(r_{\vec{x}})$. 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 $r_{\vec{x}}$, and the rotor sequences $\rho_{\vec{x}}$ for all vertices $\vec{x} \in \mathbb{Z}^d$. 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 $f(\vec{x},t) \colon \mathbb{Z}^d \times \mathbb{N}_0 \to \mathbb{N}_0$ be the number of tokens at vertex $\vec{x}$ after t steps of the (deterministic) rotor-router model, and let $\mathbb{E}(\vec{x},t) \colon \mathbb{Z}^d \times \mathbb{N}_0 \to \mathbb{R}^+$ denote the expected number of tokens after t steps of a random walk with the same starting configuration $f(\vec{x},0)$. To compute $\mathbb{E}(\vec{x},t)$ we determine for each $\vec{y} \in \mathbb{Z}^d$ the probability that a random walk starting at $\vec{y}$ reaches $\vec{x}$ after exactly t steps and multiply the result with the number of tokens that were at $\vec{y}$. Hence,

$$\mathbb{E}(\vec{x},t) = \sum_{\vec{y} \in \mathbb{Z}^d} f(\vec{y},0) \cdot S_t^d(\vec{x} - \vec{y}). \tag{2.1}$$

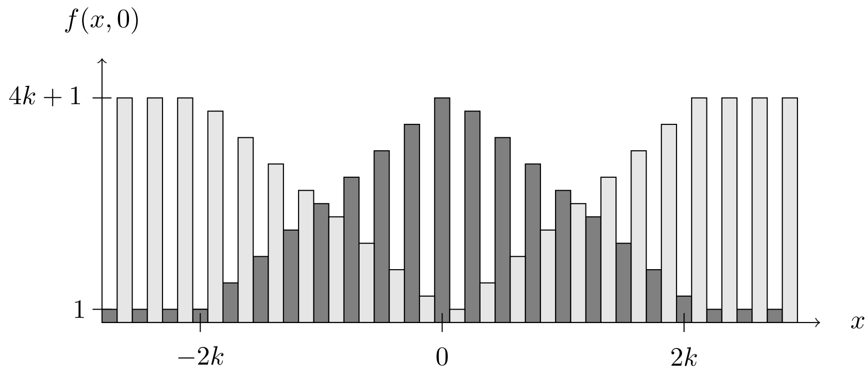

Fig. 3.1. *Illustration of the (k,d)-wedge for $k=8$ and $d=1$. The y-axis describes the number of tokens at position x. Dark colored bars show the even partition, light colored bars the odd one. This stable configuration is used to show our main result.*

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

Definition 2.1. *Let $d \geqslant 1$, and let an initial configuration $f(\vec{x},0)$ for all $\vec{x} \in \mathbb{Z}^d$ be given. We call $\Delta(\vec{x},t) = |f(\vec{x},t) - \mathbb{E}(\vec{x},t)|$ the* single vertex discrepancy at $\vec{x}$ after t steps. *Then, we define the* single vertex discrepancy Δ_d *as*

$$\Delta_d := \sup_{\vec{x}\in\mathbb{Z}^d, t\in\mathbb{N}} \Delta(\vec{x},t). \tag{2.2}$$

3. Stable configuration of the rotor-router model. According to Theorem 1.1, 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. There are several ways to model this problem. We consider the following setting most suitable for our work. The (k,d)-wedge is illustrated in Figure 3.1 and formally defined as follows.

Definition 3.1. *Let $k,d \in \mathbb{N}$ be given, where k adjusts the vertex discrepancy. The rotor direction of vertex $\vec{x}$ at time t will be denoted by $r(\vec{x},t)\colon \mathbb{Z}^d \times \mathbb{N}_0 \to E_{2d}$. We define the* (k,d)-wedge*, a starting configuration of the rotor-router model, as follows. For even vertices $\vec{x}$ with $|\vec{x}| \sim 0$, we set*

$$f(\vec{x},0) := F_0(|\vec{x}|) := \begin{cases} d\cdot(4k+1+2|\vec{x}|) & \text{if } |\vec{x}| \in [-2k,0]\,, \\ d\cdot(4k+3-2|\vec{x}|) & \text{if } |\vec{x}| \in [1,2k]\,, \\ d & \text{otherwise}, \end{cases}$$

$$r(\vec{x},0) := R_0(|\vec{x}|) := \begin{cases} -\vec{e}_1 & \text{if } |\vec{x}| \in [1,2k]\,, \\ \vec{e}_1 & \text{otherwise}. \end{cases}$$

For odd vertices $\vec{x}$ with $|\vec{x}| \sim 1$, we set

$$f(\vec{x},0) := F_1(|\vec{x}|) := \begin{cases} d\cdot(1-2|\vec{x}|) & \text{if } |\vec{x}|\in[-2k,0]\,,\\ d\cdot(2|\vec{x}|-1) & \text{if } |\vec{x}|\in[1,2k]\,,\\ d\cdot(4k+1) & \text{otherwise,}\end{cases}$$

$$r(\vec{x},0) := R_1(|\vec{x}|) := \begin{cases} -\vec{e}_1 & \text{if } |\vec{x}|\in[-2k,-1]\,,\\ \vec{e}_1 & \text{otherwise.}\end{cases}$$

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

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 $g\colon \mathbb{Z}^d\times E_{2d}\times E_{2d}\times\mathbb{N}\to\mathbb{N}$, where $g(\vec{x},\pm\vec{e}_i,\pm\vec{e}_j,t)$ denotes the number of tokens that vertex $\vec{x}$ receives from vertex $\vec{x}\pm\vec{e}_i$ at time t when $r(\vec{x}\pm\vec{e}_i,t)=\pm\vec{e}_j$. Therefore,

$$g(\vec{x},\vec{e},\vec{h},t) = \begin{cases} \lfloor\frac{f(\vec{x}+\vec{e},t)-d}{2d}\rfloor & \text{if } \operatorname{sgn}(\vec{e})=\operatorname{sgn}(\vec{h}),\\ \lfloor\frac{f(\vec{x}+\vec{e},t)+d}{2d}\rfloor & \text{otherwise,}\end{cases} \tag{3.1}$$

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

$$f(\vec{x},t+1) = \sum_{i=1}^{d} g(\vec{x},\vec{e}_i,r(\vec{x}+\vec{e}_i,t),t) + \sum_{i=1}^{d} g(\vec{x},-\vec{e}_i,r(\vec{x}-\vec{e}_i,t),t), \tag{3.2}$$

which results from summing up the number of tokens that the neighbors of $\vec{x}$ pass to $\vec{x}$ at time step t.

If the rotor-router model is initialized with the (k,d)-wedge, the number of tokens at a vertex $\vec{x}$ only depends on $|\vec{x}|$ at all times t. We can thus extend the definition of f to $f(|\vec{x}|,t)$. Consequently, we have $f(\vec{x},0)=f(|\vec{x}|,0)$ and therefore $f(\vec{x}\pm e_1,0)=f(|\vec{x}|\pm1,0)$. The same holds for $r(\vec{x},0)$. The definition of g in (3.1) can in this case be extended to $g(|\vec{x}|,\pm1,\pm\vec{e}_1,0)$, and we can simplify (3.2) to

$$\begin{aligned} f(|\vec{x}|,1) &= \sum_{i=1}^{d} g(|\vec{x}|,1,r(|\vec{x}|+1,0),0) + \sum_{i=1}^{d} g(|\vec{x}|,-1,r(|\vec{x}|-1,0),0)\\ &= d\cdot\big(g(|\vec{x}|,1,r(|\vec{x}|+1,0),0) + g(|\vec{x}|,-1,r(|\vec{x}|-1,0),0)\big). \end{aligned} \tag{3.3}$$

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

Lemma 3.2. *Given a (k,d)-wedge, it holds that*

$$r(\vec{x},1) = -r(\vec{x},0) \quad \text{and} \quad f(\vec{x},1) = \begin{cases} F_1(|\vec{x}|) & \text{if } |\vec{x}|\sim 0,\\ F_0(|\vec{x}|) & \text{if } |\vec{x}|\sim 1.\end{cases}$$

Lemma 3.3. *Given a (k,d)-wedge, it holds that $r(\vec{x},2)=r(\vec{x},0)$ and $f(\vec{x},2)=f(\vec{x},0)$.*

Lemma 3.2 states that the configuration of the rotor-router model after one step is again the (k,d)-wedge, except that it is reflected in the origin (reflecting all rotors at the same time) and then shifted by $\vec{e}_1$ to the right. Furthermore, all rotors point in the opposite direction. By the same intuition, the next step undoes these changes

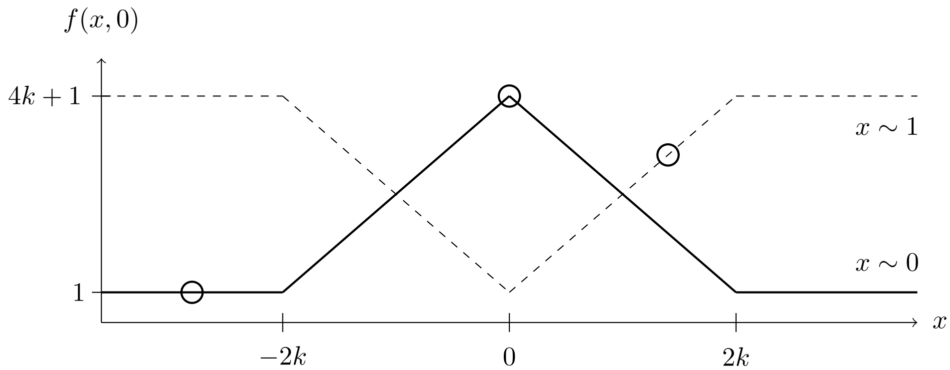

FIG. 3.2. *Illustration of the $(k, 1)$-wedge. The y-axis describes the number of tokens at position x. Circles mark the proof cases that our exemplary proof covers.*

and the configuration returns to the (k, d)-wedge after two steps, which is shown by Lemma 3.3.

These statements can be proven by a case analysis of (3.3). 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 into 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 provided an exemplary proof for three cases and used the automated prover Isabelle/HOL [20] for the remaining cases. Our code can be found in the supplementary section SM1. 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 rigorous 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. In particular, the open-source nature of the Isabelle kernel and the libraries our proof draws upon means that, as with human-readable proofs, the argument can continue to be scrutinized in perpetuity. Figure 3.2 shows which proof cases were covered in the exemplary proof.

Proof. We begin with the proof for the case $|\vec{x}| = 0$ and $t = 1$. By (3.3),

$$\begin{aligned} f(0,1) &= d \cdot (g(0,1,r(1,0),0) + g(0,-1,r(-1,0),0)) \\ &= d \cdot (g(0,1,1,0) + g(0,-1,-1,0)) \\ &= d \cdot \frac{1}{2d}(f(1,0) - d + f(-1,0) - d) \\ &= \frac{1}{2}(d \cdot (2 \cdot 1 - 1) - d + d \cdot (1 - 2 \cdot (-1)) - d) \\ &= d = F_1(0). \end{aligned}$$

This agrees with the statement.

Additionally, we present the proof for the case $|\vec{x}| \in \mathbb{Z} \setminus [-2k-1, 2k+1]$, with

$|\vec{x}| \sim 0$ and $t = 1$. By (3.3),

$$\begin{aligned} f(|\vec{x}|, 1) &= d \cdot (g(|\vec{x}|, 1, r(|\vec{x}| + 1, 0), 0) + g(|\vec{x}|, -1, r(|\vec{x}| - 1, 0), 0)) \\ &= d \cdot (g(|\vec{x}|, 1, 1, 0) + g(|\vec{x}|, -1, 1, 0)) \\ &= d \cdot \frac{1}{2d}(f(|\vec{x}| + 1, 0) - d + f(|\vec{x}| - 1, 0) + d) \\ &= \frac{1}{2}(d \cdot (4k + 1) + d \cdot (4k + 1)) \\ &= d \cdot (4k + 1) = F_1(|\vec{x}|). \end{aligned}$$

This agrees with the statement. Finally, we present the proof for the case $|\vec{x}| \in (1, 2k)$, with $|\vec{x}| \sim 1$ and $t = 1$. By (3.3),

$$\begin{aligned} f(|\vec{x}|, 1) &= d \cdot (g(|\vec{x}|, 1, r(|\vec{x}| + 1, 0), 0) + g(|\vec{x}|, -1, r(|\vec{x}| - 1, 0), 0)) \\ &= d \cdot (g(|\vec{x}|, 1, -1, 0) + g(|\vec{x}|, -1, -1, 0)) \\ &= d \cdot \frac{1}{2d}(f(|\vec{x}| + 1, 0) + d + f(|\vec{x}| - 1, 0) - d) \\ &= \frac{1}{2}(d \cdot (4k + 3 - 2(|\vec{x}| + 1)) + d \cdot (4k + 3 - 2(|\vec{x}| - 1))) \\ &= d \cdot (4k + 3 - 2|\vec{x}|) = F_0(|\vec{x}|). \end{aligned}$$

This agrees with the statement. All other cases can be shown similarly. Supplementary section SM1 presents an automated proof of all cases. ∎

3.1. 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 $\vec{x}$ 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. In particular, the two processes deviate because 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 determine the resulting 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 3.4. *If the rotor-router model is initialized with the (k, d)-wedge, we have*

$$\lim_{t \to \infty} \Delta(0, t) = 4dk.$$

Proof. Recall that $f(0, t)$ describes the number of tokens at $\vec{x} = 0$ when the rotor-router model is run, whereas $\mathbb{E}(0, t)$ describes the expected number of tokens at $\vec{x} = 0$ for the random walk after t steps. By Definition 2.1,

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

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 (2.1),

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

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

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

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

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

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 $\mathbb{E}'(\vec{x}, t)$ to denote the expected number of tokens in this random walk, and we initialize $\mathbb{E}'(\vec{x}, 0)$ again with the (k, d)-wedge. Note that $\mathbb{E}'(\vec{x}, t) = \mathbb{E}'(|\vec{x}|, t)$ again only depends on $|\vec{x}|$ and t. By showing $\mathbb{E}'(\vec{x}, t) = \mathbb{E}(\vec{x}, t)$ we can analyze a 1-dimensional random walk and directly obtain results for d-dimensional random walks.

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

$$\begin{aligned} \mathbb{E}(\vec{x}, t) &= \sum_{\vec{y} \in \mathbb{Z}^d} \mathbb{E}(\vec{y}, t-1) \cdot S_1^d(\vec{x} - \vec{y}) \qquad (3.4) \\ &= \sum_{\substack{\vec{y} \in \mathbb{Z}^d \\ |\vec{y}| = |\vec{x}| + 1}} \mathbb{E}'(|\vec{y}|, t-1) \cdot S_1^d(\vec{x} - \vec{y}) + \sum_{\substack{\vec{y} \in \mathbb{Z}^d \\ |\vec{y}| = |\vec{x}| - 1}} \mathbb{E}'(|\vec{y}|, t-1) \cdot S_1^d(\vec{x} - \vec{y}) \\ &= \mathbb{E}'(|\vec{x}| + 1, t-1) \cdot \frac{1}{2} + \mathbb{E}'(|\vec{x}| - 1, t-1) \cdot \frac{1}{2} \\ &= \mathbb{E}'(|\vec{x}|, t) = \mathbb{E}'(\vec{x}, t), \qquad (3.5) \end{aligned}$$

where (3.4) and (3.5) 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 $f(\vec{x}, t) = d$ 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 $\mathbb{E}_S(0, t)$:

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

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

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

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

The number of expected tokens that start in I_1 and end at the origin after t steps will be approximated using the upper bound $\binom{t}{t/2} \leqslant \sqrt{\frac{2}{\pi t}} \cdot 2^t$ [22].

Then, $\mathbb{E}_{I_1}$ can be estimated the following way:

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

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 obtaining $\mathbb{E}(0,t) \leqslant d + \sqrt{\frac{2}{\pi t}} \cdot d \cdot (2k+1)^2$. This results in a discrepancy of

$$|f(0,t) - \mathbb{E}(0,t)| \geqslant \max \left\{ 0, 4dk - \sqrt{\frac{2}{\pi t}} \cdot d \cdot (2k+1)^2 \right\}. \tag{3.6}$$

We obtain $\Delta(0,t) \geqslant 4dk$ for $t \to \infty$. It remains to show that $\Delta(0,t) \leqslant 4dk$.

Recall that $f(\vec{x},0) \geqslant d$ for all $\vec{x}$. Thus, in the random walk process, every vertex distributes at least d tokens evenly among its neighbors, i.e., each of the $2d$ neighbors obtains at least $d/(2d)$ tokens in expectation. It follows that every vertex, in total, receives at least d tokens from its $2d$ neighbors, which it distributes evenly in the next step. With this invariant, it is easy to see that $\mathbb{E}(\vec{x},t) \geqslant d$ for all $\vec{x}$ and t. Analogously, we have $\mathbb{E}(\vec{x},t) \leqslant d(4k+1)$ for all $\vec{x}$ and t, since $f(\vec{x},0) \leqslant d(4k+1)$ for all $\vec{x}$. Additionally, due to the stability of the (k,d)-wedge (Lemma 3.3), we know that $d \leqslant f(\vec{x},t) \leqslant d(4k+1)$ holds for all $\vec{x}$ and t. Taken together, we can derive that $0 \leqslant \Delta(\vec{x},t) \leqslant 4dk$ holds for all $\vec{x}$ and t, and in particular, $\Delta(0,t) \leqslant 4dk$ for all t. ◻

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 3.3 illustrates the single vertex discrepancy in a $(k,1)$-wedge over time for $k \in \{16, 32, 64\}$.

3.2. Discrepancy within finite steps.

Lemma 3.4 shows that a discrepancy of $4dk$ can be reached if the processes are run for $t \to \infty$ steps. It is, however, possible to achieve high discrepancy using already few steps by investigating (3.6) more carefully. We show the following corollary.

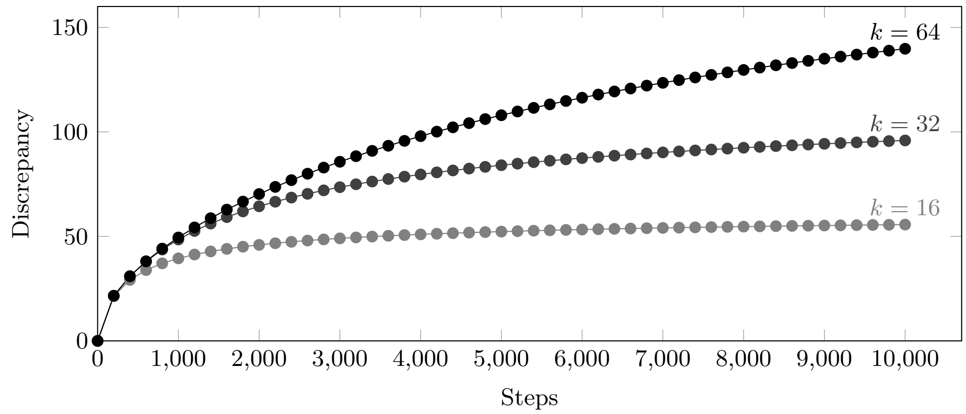

FIG. 3.3. *The simulated single vertex discrepancies for different* $(k, 1)$*-wedges. The plots show that even for small* t *and* k *a high discrepancy can be achieved. This intuition is formalized in Corollary* 3.5.

COROLLARY 3.5. *Given dimension* $d \geqslant 1$ *and a discrepancy* $\ell \in \mathbb{R}_+$*, there exists a* (k, d)*-wedge that reaches the discrepancy* ℓ *in* $t \in \mathcal{O}\left(\lceil \ell^2/d^2 \rceil\right)$ *steps using* $\mathcal{O}(\lceil 1 + \ell/d \rceil^{2d+1})$ *tokens.*

Proof. By (3.6), the number of steps that are needed to reach discrepancy ℓ with a (k, d)-wedge are

$$\ell \leqslant 4dk - \sqrt{\frac{2}{\pi t}} \cdot d \cdot (2k+1)^2,$$
$$\Leftarrow t \geqslant \frac{2}{\pi} \cdot \frac{d^2(2k+1)^4}{(4dk-\ell)^2}.$$

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 $\leqslant d \cdot (4k+1)$ tokens. Therefore, in total it needs at most $\mathcal{O}(\lceil 1 + \ell/d \rceil^{2d+1})$ tokens. ∎

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 [12]. We study the rotor-router model's similarity to the expected behavior of the random walk. It was observed and well studied that on grids tokens only differs by some small constant at all times and on each vertex [5, 8, 6]. 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] R. Aleliunas, R. M. Karp, R. J. Lipton, L. Lovász, and C. Rackoff, *Random walks, universal traversal sequences, and the complexity of maze problems*, in Proceedings of the 20th Annual Symposium on Foundations of Computer Science, IEEE, New York, 1979, pp. 218–223.

[2] E. Bampas, L. Gąsieniec, N. Hanusse, D. Ilcinkas, R. Klasing, and A. Kosowski, *Euler tour lock-in problem in the rotor-router model*, in Proceedings of the 23rd International Conference on Distributed Computing, Springer-Verlag Berlin, Heidelberg, 2009, pp. 423–435.

[3] R. D. Barve, E. F. Grove, and J. S. Vitter, *Simple randomized mergesort on parallel disks*, Parallel Comput., 23 (1997), pp. 601–631, also in Proceedings of the Eighth Annual ACM Symposium on Parallel Algorithms and Architectures, 1996, pp. 109–118.

[4] J. Cooper, B. Doerr, T. Friedrich, and J. Spencer, *Deterministic random walks on regular trees*, Random Structures Algorithms, 37 (2010), pp. 353–366, also in Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms, 2008, pp. 766–772.

[5] J. Cooper, B. Doerr, J. Spencer, and G. Tardos, *Deterministic random walks on the integers*, European J. Combin., 28 (2007), pp. 2072–2090, also in Proceedings of the Workshop on Analytic Algorithmics and Combinatorics, SIAM, Philadelphia, 2006, pp. 185–197.

[6] J. N. Cooper and J. Spencer, *Simulating a random walk with constant error*, Combin. Probab. Comput., 15 (2006), pp. 815–822.

[7] D. Dereniowski, A. Kosowski, D. Pająk, and P. Uznański, *Bounds on the cover time of parallel rotor walks*, in Proceedings of the 31st International Symposium on Theoretical Aspects of Computer Science, 2014, pp. 263–275.

[8] B. Doerr and T. Friedrich, *Deterministic random walks on the two-dimensional grid*, Combin. Probab. Comput., 18 (2009), pp. 123–144, also in Proceedings of ISAAC, 2006, pp. 474–483.

[9] B. Doerr, T. Friedrich, and T. Sauerwald, *Quasirandom rumor spreading*, ACM Trans. Algorithms, 11 (2014), 9, https://doi.org/10.1145/2650185, also in Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms, 2008, pp. 773–781.

[10] I. Dumitriu, P. Tetali, and P. Winkler, *On playing golf with two balls*, SIAM J. Discrete Math., 16 (2003), pp. 604–615, https://doi.org/10.1137/S0895480102408341.

[11] M. Dyer, A. Frieze, and R. Kannan, *A random polynomial-time algorithm for approximating the volume of convex bodies*, J. ACM, 38 (1991), pp. 1–17, also in Proceedings of the Twenty-first Annual ACM Symposium on Theory of Computing, 1989, pp. 375–381.

[12] T. Friedrich, M. Gairing, and T. Sauerwald, *Quasirandom load balancing*, SIAM J. Comput., 41 (2012), pp. 747–771, https://doi.org/10.1137/100799216, also in Proceedings of the Twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms, 2010, pp. 1620–1629.

[13] T. Friedrich, M. Katzmann, and A. Krohmer, *Unbounded discrepancy of deterministic random walks on grids*, in Proceedings of the 26th International Symposium on Algorithms and Computation,Nagoya, Japan, 2015, pp. 212–222.

[14] T. Friedrich and T. Sauerwald, *The cover time of deterministic random walks*, Electron. J. Combin., 17 (2010), Research paper 167, also in Proceedings of the 16th International Computing and Combinatorics Conference, 2010, pp. 130–139.

[15] S. Kijima, K. Koga, and K. Makino, *Deterministic random walks on finite graphs*, in 9th Meeting on Analytic Algorithmics and Combinatorics, 2012, pp. 16–25.

[16] R. Klasing, A. Kosowski, D. Pająk, and T. Sauerwald, *The multi-agent rotor-router on the ring: A deterministic alternative to parallel random walks*, in Proceedings of the 32nd ACM Symposium on Principles of Distributed Computing, 2013, pp. 365–374.

[17] M. Kleber, *Goldbug variations*, Math. Intelligencer, 27 (2005), pp. 55–63.

[18] A. Kosowski and D. Pająk, *Does adding more agents make a difference? A case study of cover time for the rotor-router*, in 41st International Colloquium on Automata, Languages, and Programming, 2014, pp. 544–555.

[19] G. Lawler and V. Limic, *Random Walk: A Modern Introduction*, Cambridge Studies in Advanced Mathematics, Cambridge University Press, Cambridge, 2010.

[20] T. Nipkow, L. C. Paulson, and M. Wenzel, *Isabelle/HOL: A Proof Assistant for Higher-Order Logic*, Lecture Notes in Comput. Sci. 2283, Springer-Verlag, Berlin, Heidelberg, 2002.

[21] V. B. Priezzhev, D. Dhar, A. Dhar, and S. Krishnamurthy, *Eulerian walkers as a model of self-organized criticality*, Phys. Rev. Lett., 77 (1996), pp. 5079–5082.

[22] H. Robbins, *A remark on Stirling's formula*, Amer. Math. Monthly, 62 (1955), pp. 26–29.

[23] I. A. Wagner, M. Lindenbaum, and A. M. Bruckstein, *Distributed covering by ant-robots using evaporating traces*, IEEE Trans. Robot. Autom., 15 (1999), pp. 918–933.

[24] V. Yanovski, I. A. Wagner, and A. M. Bruckstein, *A distributed ant algorithm for efficiently patrolling a network*, Algorithmica, 37 (2003), pp. 165–186.