

Analyzing Disturbed Diffusion on Networks^{*}

Henning Meyerhenke and Thomas Sauerwald

Universität Paderborn
Fakultät für Elektrotechnik, Informatik und Mathematik
Fürstenallee 11, D-33102 Paderborn, Germany
email: {henningm|sauerwal}@upb.de

Abstract. This work provides the first detailed investigation of the disturbed diffusion scheme FOS/C introduced in [17] as a type of diffusion distance measure within a graph partitioning framework related to Lloyd's k -means algorithm [14]. After outlining connections to distance measures proposed in machine learning, we show that FOS/C can be related to random walks despite its disturbance. Its convergence properties regarding load distribution and edge flow characterization are examined on two different graph classes, namely torus graphs and distance-transitive graphs (including hypercubes), representatives of which are frequently used as interconnection networks.

Keywords: Disturbed diffusion, Diffusion distance, Random walks.

1 Introduction

Diffusive processes can be used to model a large variety of important transport phenomena arising in such diverse areas as heat flow, particle motion, and the spread of diseases. In computer science one has studied diffusion in graphs as one of the major tools for balancing the load in parallel computations [5], because it requires only local communication between neighboring processors. Equally important, the migrating flow computed by diffusion is $\|\cdot\|_2$ -optimal [6].

Recently, disturbed diffusion schemes have been developed as part of a graph partitioning heuristic [17,20]. Applied within a learning framework optimizing the shape of the partitions, disturbed diffusion is responsible for identifying densely connected regions in the graph. As partitions are placed such that their centers are located within these dense regions, this heuristic yields partitions with few boundary nodes. This is desirable particularly in scientific computing, where the boundary nodes model the communication within parallel numerical solvers. The disturbed diffusion scheme and the algorithm employing it are described in more detail in Sections 2.1 and 2.2, respectively.

^{*} This work is partially supported by German Science Foundation (DFG) Research Training Group GK-693 of the Paderborn Institute for Scientific Computation (PaSCo) and by Integrated Project IST-15964 "Algorithmic Principles for Building Efficient Overlay Computers" (AEOLUS) of the European Union.

While the connection between diffusion and random walks on graphs is well-known (see, e.g., [15]), the relation of disturbed diffusion the way considered here to random walks has not been explored yet. In Section 3 we thus show that random walk analysis can be applied despite the disturbance in the diffusion scheme. Before that, we draw connections to machine learning applications that employ distance measures based on diffusion and random walks in Section 2.3.

Using random walk theory, we analyze the load distribution and the edge flow induced by FOS/C in its convergence state on torus graphs in Section 4. Although random walks on infinite and finite tori have been investigated before (cf. Pólya’s problem [7] and [8]), our monotonicity result on the torus provides an important theoretical property, to the best of our knowledge previously unknown. It is of high relevance for the graph partitioning heuristic, since the torus corresponds to structured grids that stem from the discretization of numerical simulation domains with cyclic boundary conditions. A simple characterization of the convergence flow by shortest paths is shown not to hold on the two-dimensional torus in general. Interestingly, this characterization is true on distance-transitive graphs, as shown in Section 5, which is supplemented by a more rigorous result for the hypercube, a very important representative of distance-transitive graphs.

These insights provide a better understanding of FOS/C, its properties within the graph partitioning framework, and its connection to similar distance measures in machine learning. They are expected to improve the partitioning heuristic in theory and practice and its applicability for graph clustering and related applications.

2 Disturbed Diffusion, Graph Partitioning, and Diffusion Distances

2.1 Disturbed Diffusion: FOS/C

Diffusion is one method for iteratively balancing the load on vertices of a graph by performing load exchanges only between neighboring vertices [26]. The idea of the FOS/C algorithm (C for constant drain) is to modify the first order diffusion scheme FOS [5] by letting some of the load on each vertex drain away after each diffusion step. The total drain of the graph is then sent back to some specified source vertex s , before the next iteration begins. For this algorithm the underlying graph has to be connected, undirected, and loop-free. Additionally, we assume it to be unweighted and simple throughout this paper.

Definition 1. (comp. [17]) *Given a graph $G = (V, E)$ with n nodes, a specified source vertex s , and constants $0 < \alpha \leq (\deg(G) + 1)^{-1}$ and $\delta > 0$. Let the initial load vector $w^{(0)}$ and the disturbing drain vector d be defined as follows:*

$$w_v^{(0)} = \begin{cases} n & v = s \\ 0 & \text{otherwise} \end{cases} \quad d_v = \begin{cases} \delta(n-1) & v = s \\ -\delta & \text{otherwise} \end{cases}$$

The FOS/C diffusion scheme performs the following operations in each iteration:

$$f_{e=(u,v)}^{(t)} = \alpha(w_u^{(t)} - w_v^{(t)}), \quad w_v^{(t+1)} = w_v^{(t)} + d_v + \sum_{e=(*,v)} f_e^{(t)}.$$

This can be written in matrix-vector notation as $w^{(t+1)} = \mathbf{M}w^{(t)} + d$, where $\mathbf{M} = \mathbf{I} - \alpha\mathbf{L}$ is the stochastic diffusion matrix of G (and \mathbf{L} its Laplacian [10]). It is shown in [17] that FOS/C converges for any d that preserves the total load amount in every iteration, i.e., $d \perp (1, \dots, 1)^T$. Moreover, in this case the convergence load vector w can be computed by first solving the linear system $\mathbf{L}w = d$ and then normalizing w such that the total load is n again. The entries of this vector can then be interpreted as the diffusion distances between s and the other nodes.

Comparing this notation to [6] and [11], it is clear that the convergence state of FOS/C is equivalent to the following flow problem: Find the $\|\cdot\|_2$ -minimal flow from the producing source s sending the respective load amount δ to all other vertices in the graph, which act as δ -consuming sinks. One therefore knows that s always has the highest load.

Remark 1. Using Lemma 4 of [6], it follows that $f = \mathbf{A}^T w$ (with \mathbf{A} being the incidence matrix of G [10, p. 58]) is the $\|\cdot\|_2$ -minimal flow via the edges of G induced by the flow problem equivalent to FOS/C. Hence: $f_{e=(u,v)} = w_u - w_v$.

Furthermore, it holds for every path between any $u, v \in V$ that the sum of the load differences $\sum_{i=0}^{l-1} (w_{v_i} - w_{v_{i+1}})$ on the path edges $e_i = (v_i, v_{i+1})$ is equal to $w_u - w_v$ (with $u = v_0$ and $v = v_l$).

The following proposition states a basic monotonicity result that holds on any graph, whereas stricter results will be proven in the forthcoming sections.

Proposition 1. *Let the graph $G = (V, E)$ and the load vector w be given. Then for each vertex $v \in V$ there is a path $(v = v_0, v_1, \dots, v_l = s)$ with $(v_i, v_{i+1}) \in E$ such that $w_{v_i} < w_{v_{i+1}}, 0 \leq i < l$.*

2.2 FOS/C for Graph Partitioning

The graph partitioning framework in which FOS/C is applied transfers Lloyd's algorithm [14] well-known from k -means-type cluster analysis and least square quantization to graphs. Starting with k (the number of partitions) randomly chosen center vertices, all remaining nodes are assigned to the closest center based on FOS/C. This means that we solve one FOS/C diffusion problem per partition (its center acts as source s) and assign each vertex to the partition which sends the highest load in the convergence state. After that, each partition computes its new center (based on a similar FOS/C problem again) for the next iteration. This can be repeated until a stable state, where the movement of all centers is small enough, is reached. For a detailed discussion of this iterative algorithm called Bubble-FOS/C the reader is referred to [17].

2.3 Diffusion Distances in Graphs

One can view FOS/C as a means to determine the distance from each vertex to the different center vertices within Bubble-FOS/C (hence, ordinary FOS is not applicable, because it converges to a completely balanced load situation), where this distance reflects how well-connected the two vertices are (comp. [19] and [7, p. 99f.]). Thus, it is able to identify dense regions of the graph. A similar idea is pursued by other works that make use of distance measures based on random walks and diffusion. They have mostly been developed for machine learning, namely, clustering of point sets and graphs [18,19,23,25,27], image segmentation [16], and dimensionality reduction [4]. However, their approaches rely on very expensive matrix operations, amongst others computation of matrix powers [23,25], eigenvectors of a kernel matrix [4,16,18], or the pseudoinverse of the graph's Laplacian [19,27]. This mostly aims at providing a distance between every pair of nodes.

Yet, this is not necessary for Lloyd's algorithm, because distance computations are relative to the current centers and the determination of the new partition centers can also be replaced by a slightly modified FOS/C operation, as mentioned above. The sparse linear system $\mathbf{L}w = d$, where w can be seen as the result of the pseudoinverse's impact on the drain vector, can be solved with $\mathcal{O}(n^{3/2})$ and $\mathcal{O}(n^{4/3})$ operations for typical $2D$ and $3D$ finite-element graphs, respectively, using the conjugate gradient algorithm [21]. This can even be enhanced by (algebraic) multigrid methods [24], which have linear time complexity when implemented with care. Note that only a constant number of calls to FOS/C are sufficient in practice. Thus, this approach is faster (unless distances between every pair of nodes are necessary in a different setting) than the related methods, which all require at least $\mathcal{O}(n^2)$ operations in the general case.

3 Relating FOS/C to Random Walks

In order to examine the relationship between disturbed diffusion and random walks, we expand the original definition of FOS/C and obtain

$$w^{(t+1)} = \mathbf{M}^{t+1}w^{(0)} + (\mathbf{I} + \mathbf{M}^1 + \dots + \mathbf{M}^t)d.$$

Note that the doubly stochastic diffusion matrix \mathbf{M} of G in the classical FOS diffusion scheme can be viewed as the transition matrix of a random walk [15] on $V(G)$, i.e., $\mathbf{M}_{u,v}$ denotes the probability for a random walker located in node u to move to node v in the next timestep. Despite its disturbance, a similar connection holds for FOS/C, since its load differences in the convergence state (a.k.a. *stationary distribution* in random walk theory) can be expressed as scaled differences of hitting times, as shown below. In the following let $X_u^{(t)}$ be the random variable representing the node visited in timestep t by a random walker starting in u in timestep 0.

Definition 2. Let the balanced distribution vector be $\pi = (\frac{1}{n}, \dots, \frac{1}{n})^T$ and let τ_u be defined as $\tau_u := \min\{t \geq 0 : X_u^{(t)} = s\}$ for any $u \in V$. Then, the hitting time H is defined as $H[u, s] := \mathbb{E}[\tau_u]$.

Theorem 1. In the convergence state it holds for two nodes $u, v \in V$ not necessarily distinct from s

$$w_u - w_v = \lim_{t \rightarrow \infty} n\delta \left(\sum_{i=0}^t \mathbf{M}_{u,s}^i - \sum_{i=0}^t \mathbf{M}_{v,s}^i \right) = \delta(H[v, s] - H[u, s]).$$

Proof. We denote the component corresponding to node u in a vector w by $[w]_u$ and assume that the nodes are ordered in such a way that the source node is the first one. Then some rearranging of the FOS/C iteration scheme yields

$$\begin{aligned} [w^{(t+1)}]_u &= [\mathbf{M}^{t+1}w^{(0)}]_u + [(\mathbf{I} + \mathbf{M}^1 + \dots + \mathbf{M}^t) \cdot (\delta(n-1), -\delta, \dots, -\delta)^T]_u \\ &= [\mathbf{M}^{t+1}w^{(0)}]_u + \sum_{i=0}^t (\delta(|V| - 1))\mathbf{M}_{u,s}^i + \sum_{i=0}^t \sum_{v \in V, v \neq s} (-\delta)\mathbf{M}_{u,v}^i \\ &= [\mathbf{M}^{t+1}w^{(0)}]_u + n\delta \sum_{i=0}^t \mathbf{M}_{u,s}^i - (t+1)\delta. \end{aligned}$$

As $\mathbf{M}^{t+1}w^{(0)}$ converges towards the balanced load distribution [5], we only have to consider $\lim_{t \rightarrow \infty} \sum_{i=0}^t (\mathbf{M}_{u,s}^i - \mathbf{M}_{v,s}^i)$. By a result of [12, p. 79] it holds that $H[u, s] = (-\sum_{k=1}^{\infty} M_{u,s}^k + \sum_{k=1}^{\infty} (1/n) + Z_{s,s}) \cdot n$, where Z is the so-called fundamental matrix. Now, subtracting and dividing by n yields the desired result.

4 FOS/C on the Torus

In this section we analyze two properties of FOS/C on torus graphs in the convergence state, namely, its edge flow and the corresponding load distribution.

Definition 3. The k -dimensional torus $T[d_1, \dots, d_k] = (V, E)$ is defined as:

$$\begin{aligned} V &= \{(u_1, \dots, u_k) \mid 0 \leq u_\nu \leq d_\nu - 1 \text{ for } 1 \leq \nu \leq k\} \text{ and} \\ E &= \{ \{(u_1, \dots, u_k), (v_1, \dots, v_k)\} \mid \exists 1 \leq \mu \leq k \\ &\quad \text{with } v_\mu = (u_\mu + 1) \bmod d_\mu \text{ and } u_\nu = v_\nu \text{ for } \nu \neq \mu \}. \end{aligned}$$

Torus graphs are very important in theory [13] and practice [22], e.g., because they have bounded degree, are regular and vertex-transitive¹, and correspond to the structure of numerical simulation problems that decompose their domain by structured grids with cyclic boundary conditions. Note that the load distribution on a torus and a grid graph are equal if their d_i are all odd and s is located at the center of the graphs, because then there is no flow via the wraparound edges of the torus.

¹ A graph $G = (V, E)$ is vertex-transitive if for any two distinct vertices of V there is an automorphism mapping one to the other.

Since the number of shortest paths from a source s to another vertex u does not depend on its distance to s alone, the following flow distribution among the shortest paths is not optimal on the torus in general. As we will see later, this optimality holds for graphs that are distance-transitive, an even stronger symmetry property than vertex-transitivity.

Definition 4. *Consider the flow problem where s sends a load amount of δ to every other vertex of G , which acts as a δ -consuming sink. If the flow is distributed such that for all $v \in V \setminus \{s\}$ the same flow amount is routed on every (not necessarily edge-disjoint) shortest path from s to v , we call this the uniform flow distribution.*

Proposition 2. *The uniform flow distribution on the 2D torus yields the $\|\cdot\|_2$ -minimal flow for $d_1 = d_2 \in \{2, 3, 5\}$, but not for odd $d_1 = d_2 \geq 7$.*

Intuitively, the reason is that near the diagonal there are more shortest paths than on an axis and thus, by rerouting some of the uniform flow towards the diagonal, the costs can be reduced.

In the remainder of this section we exploit the simple structure and symmetries of the torus to show monotonicity w.r.t. the FOS/C convergence load distribution. Since we are only interested in the convergence state, we will set $\alpha = (\deg(G) + 1)^{-1}$, so that all entries of the diffusion matrix \mathbf{M} are either 0 or α . This is a usual choice for transition matrices in random walk theory.

Now consider an arbitrary k -dimensional torus $T[d_1, \dots, d_k]$. Each vertex u can be uniquely represented as a k -dimensional vector $u = (u_1, \dots, u_k), \forall i \in \{1, \dots, k\} : 0 \leq u_i < d_i$. Since any torus is vertex-transitive, we assume w.l.o.g. that the source node is the zero-vector. Denote by $\mathbf{e}_i = (0, \dots, 0, 1, 0, \dots, 0)$ the unit-vector containing exactly one 1, namely in the i -th component. Note that all edges correspond to the addition (or subtraction) of some \mathbf{e}_i , where we always assume that the i -th component is meant to be modulo d_i . It is also easy to see that the distance between two nodes (vectors) is given by $\text{dist}(u, v) = \sum_{i=1}^k \min\{|u_i - v_i|, d_i - |u_i - v_i|\}$.

Let u, v, s be pairwise distinct nodes such that $\text{dist}(u, s) = \text{dist}(v, s) - 1$ and u and v are adjacent, i.e., there exists a shortest path from s to v via u . Assume w.l.o.g. that u and v are adjacent along the j -th dimension: $v = u + \mathbf{e}_j$, so that

$$\forall i \in \{1, \dots, k\}, i \neq j : \text{dist}(v, s) - \text{dist}(v, s \pm \mathbf{e}_i) = \text{dist}(u, s) - \text{dist}(u, s \pm \mathbf{e}_i),$$

implying the existence of a shortest path from $s \pm \mathbf{e}_i$ to v via $u \forall i \neq j$.

For vertex-transitive graphs G , all $\varphi \in \text{Aut}(G)$, and all timesteps t we have $\mathbf{M}_{u,v}^t = \mathbf{M}_{\varphi(u),\varphi(v)}^t$ [2, p. 151]. Using this and the automorphisms of the next lemma, we prove the following theorem, which may be of independent interest for random walks in general.

Lemma 1. *The following functions are automorphisms for all $i \in \{1, \dots, k\}$: $\psi_i : u \mapsto u + \mathbf{e}_i$, $\varphi_i : u \mapsto u + (d_i - 2u_i)\mathbf{e}_i$, and $\sigma_i : u \mapsto u + (d_i - 1 - 2u_i)\mathbf{e}_i$.*

Theorem 2. Let $T[d_1, \dots, d_k] = (V, E)$, k arbitrary, be a torus graph. For $\alpha = (\deg(G) + 1)^{-1}$ and all adjacent nodes $u, v \in V$ distinct from s with $\text{dist}(u, s) = \text{dist}(v, s) - 1$ it holds

$$\forall t \in \mathbb{N}_0 : \mathbf{M}_{u,s}^t \geq \mathbf{M}_{v,s}^t.$$

Proof. We will prove the statement by induction on the number of timesteps t . Obviously, the claim is true for $t = 0$. By the Chapman-Kolmogorov equation, see e.g. [9], we have

$$\mathbf{M}_{u,s}^t = \frac{1}{\Delta + 1} \left(\mathbf{M}_{u,s}^{t-1} + \sum_{i \in \{1, \dots, k\}} \mathbf{M}_{u, s + \mathbf{e}_i}^{t-1} + \sum_{i \in \{1, \dots, k\}} \mathbf{M}_{u, s - \mathbf{e}_i}^{t-1} \right). \quad (1)$$

Obviously, the same equation holds also for $M_{v,s}^t$. Our strategy is now to find for any summand in $M_{v,s}^t$ a proper summand in $M_{u,s}^t$ which is not smaller by using the induction hypothesis for $t - 1$. Of course, if this is done bijectively, we have shown that $M_{u,s}^t \geq M_{v,s}^t$. To proceed, we divide this proof into two cases.

1. Case $u_j = 0$: By Lemma 1 we have

$$\begin{aligned} \mathbf{M}_{u,s}^{t-1} &= \mathbf{M}_{\psi_j(u), \psi_j(s)}^{t-1} = \mathbf{M}_{v, s + \mathbf{e}_j}^{t-1}. \\ \mathbf{M}_{u, s + \mathbf{e}_j}^{t-1} &= \mathbf{M}_{\varphi_j(u), \varphi_j(s + \mathbf{e}_j)}^{t-1} = \mathbf{M}_{u, s - \mathbf{e}_j}^{t-1} = \mathbf{M}_{\psi_j(u), \psi_j(s - \mathbf{e}_j)}^{t-1} = \mathbf{M}_{v, s}^{t-1}. \end{aligned}$$

To show $\mathbf{M}_{u, s - \mathbf{e}_j}^{t-1} \geq \mathbf{M}_{v, s - \mathbf{e}_j}^{t-1}$, we have to distinguish the following cases:

(a) Ignoring the trivial case $d_j = 2$, we now consider the case where $d_j = 3$:

$$\mathbf{M}_{u, s - \mathbf{e}_j}^{t-1} = \mathbf{M}_{u, s + \mathbf{e}_j}^{t-1} \stackrel{\psi_j^{-1}}{=} \mathbf{M}_{u - 2\mathbf{e}_j, s - \mathbf{e}_j}^{t-1} = \mathbf{M}_{v, s - \mathbf{e}_j}^{t-1}.$$

(b) $d_j \geq 4$: Then, $\text{dist}(v, s - \mathbf{e}_j) = \text{dist}(v, s) + 1$, implying the existence of a shortest path from v to $s - \mathbf{e}_j$ via u . Due to vertex-transitivity there exists an automorphism which maps $s - \mathbf{e}_j$ onto s and we can apply the induction hypothesis to conclude $\mathbf{M}_{u, s - \mathbf{e}_j}^{t-1} \geq \mathbf{M}_{v, s - \mathbf{e}_j}^{t-1}$.

Recall that for all $i \in \{1, \dots, k\}, i \neq j$, there exists a shortest path from v to $s \pm \mathbf{e}_i$ via u , so that we can again conclude inductively that $\mathbf{M}_{u, s \pm \mathbf{e}_i}^{t-1} \geq \mathbf{M}_{v, s \pm \mathbf{e}_i}^{t-1}$. With Equation (1) and its analogon for v the claim $\mathbf{M}_{u,s}^t \geq \mathbf{M}_{v,s}^t$ follows.

2. Case $u_j \neq 0$: One distinguishes two subcases by the parity of u_j and uses similar methods as before to prove this case. It is therefore omitted due to space constraints. \square

Note that one can show with a modified three-dimensional hypercube as a counterexample that this monotonicity does not hold for all vertex-transitive graphs in all timesteps. Furthermore, the general result $\mathbf{M}_{u,u}^{2t} \geq \mathbf{M}_{u,v}^{2t}$ for random walks without loops on vertex-transitive graphs can be found in [2, p. 150], which is improved significantly by our last theorem on torus graphs. As one can prove by induction, on the torus the source vertex is the unique node with the highest load in all timesteps due to the choice of α and the back-flow of the drain. Thus, by combining Theorems 1 and 2, one can derive the following corollary for any pair of vertices.

Corollary 1. *On any torus graph $T = (V, E)$ it holds for all $u, v \in V$: $\forall t < \text{dist}(u, s) : w_u^{(t)} = w_v^{(t)}$, $\forall t \in \{\text{dist}(u, s), \dots, \infty\} : w_u^{(t)} > w_v^{(t)}$.*

Using this monotonicity and the symmetry properties of the torus, it is easy (but rather technical) to show that Bubble-FOS/C produces connected partitions on this graph class, which is desirable in some applications.

5 FOS/C on Distance-transitive Graphs

We have seen that the convergence flow does not equal the uniform flow distribution on the torus, despite its symmetry. Yet, in this section we show that this equality holds if the symmetry is extended to distance-transitivity.

Definition 5. *[3, p. 118] A graph $G = (V, E)$ is distance-transitive if, for all vertices $u, v, x, y \in V$ such that $\text{dist}(u, v) = \text{dist}(x, y)$, there exists an automorphism φ for which $\varphi(u) = x$ and $\varphi(v) = y$.*

One important subclass of distance-transitive graphs are Hamming graphs, which occur frequently in coding theory [1, p. 46]. A very well-known representative is the hypercube network [13]. It is not difficult to show that distance-transitive graphs $G = (V, E)$ have a level structure w.r.t. to an arbitrary $s \in V$, where level i consists of the vertex set $L_i := \{v \in V \mid \text{dist}(v, s) = i\}$ and Λ denotes the number of such levels. For the k -dimensional hypercube $Q(k)$, for instance, we have $\Lambda = k + 1$.

Now, the results of this section can be derived by means of this level structure and the aforementioned equivalence of FOS/C to a $\|\cdot\|_2$ -minimal flow problem.

Proposition 3. *Let G be a distance-transitive graph. Then, $w_u^{(t)} = w_v^{(t)}$ holds for all vertices u, v with the same graph distance to s and all timesteps $t \geq 0$.*

We know by Proposition 1 that for each vertex $v \in V \setminus \{s\}$ of an arbitrary graph there exists a path from v to s such that by traversing it the load amount increases. Now we can show that for distance-transitive graphs this property holds on *every shortest* path.

Theorem 3. *If G is distance-transitive, then for all $u, v \in V$ with $\text{dist}(u, s) < \text{dist}(v, s)$ it holds that $w_u > w_v$.*

Note that, although the order induced by the FOS/C diffusion distance corresponds to the one induced by the ordinary graph distance, the load differences across levels reflect their connectivity (see also Theorem 5). We now state the following characterization of the convergence flow.

Theorem 4. *The uniform flow distribution of Definition 4 yields the $\|\cdot\|_2$ -minimal FOS/C convergence flow on every distance-transitive graph.*

As this is not true for general tori, the following implication is not an equivalence.

Proposition 4. *If on a graph $G = (V, E)$ the uniform flow distribution is $\|\cdot\|_2$ -minimal, then for $(u, v) \in E$ and $\text{dist}(u, s) < \text{dist}(v, s)$ it holds that $w_u > w_v$.*

Due to the explicitly known structure of the hypercube we obtain:

Theorem 5. *For the k -dimensional hypercube $Q(k) = (V, E)$ the result of Theorem 3 holds in all timesteps $t \geq 0$. Also, the FOS/C convergence flow f_e on an edge $e = (u, v) \in E$ (u in level i , v in level $i+1$, $0 \leq i < \Lambda$) is $w_u - w_v = f_e = \frac{\delta}{\binom{k}{i}\binom{k-i}{i}} \cdot \sum_{l=i+1}^k \binom{k}{l}$.*

6 Conclusions

We have shown that the disturbed diffusion scheme FOS/C can be related to random walks despite its disturbance, since its load differences in the convergence state correspond to scaled differences of hitting times. Exploiting this correspondence, we have shown that load diffuses monotonically decreasing from a source vertex into the graph on torus and distance-transitive graphs. Furthermore, while the uniform flow division among shortest paths does not yield the $\|\cdot\|_2$ -minimal flow on the torus in general, it does so on distance-transitive graphs. For the hypercube, one of its highly relevant representatives, the convergence flow has been stated explicitly.

Future work includes the extension of the results to further graph classes and simple characterizations of the convergence flow as in the case of distance-transitive graphs. Naturally, different disturbed diffusion schemes and drain concepts and therefore different distance measures could be examined as well. Moreover, while connectedness of partitions can be observed in experiments and verified easily for torus and distance-transitive graphs with the results of this paper, a rigorous proof for general graphs remains an object of further investigation, likewise a convergence proof for Bubble-FOS/C on general graphs. All this aims at further improvements to the heuristic in theory and practice for graph partitioning and its extension to graph clustering.

References

1. J. Adámek. *Foundations of Coding*. J. Wiley & Sons, 1991.
2. N. Alon and J. H. Spencer. *The Probabilistic Method*. J. Wiley & Sons, 2nd edition, 2000.
3. N. Biggs. *Algebraic Graph Theory*. Cambridge University Press, 1993.
4. R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data. Parts I and II. *Proc. Natl. Academy of Sciences*, 102(21):7426–7437, 2005.
5. G. Cybenko. Dynamic load balancing for distributed memory multiprocessors. *Parallel and Distributed Computing*, 7:279–301, 1989.
6. R. Diekmann, A. Frommer, and B. Monien. Efficient schemes for nearest neighbor load balancing. *Parallel Computing*, 25(7):789–812, 1999.

7. P. G. Doyle and J. L. Snell. *Random Walks and Electric Networks*. Math. Assoc. of America, 1984.
8. R. B. Ellis. Discrete green's functions for products of regular graphs. In *AMS National Conference, invited talk, special session on Graph Theory*, 2001.
9. G. R. Grimmett and D. R. Stirzaker. *Probability and Random Processes*. Oxford University Press, second edition, 1992.
10. J. L. Gross and J. Yellen (eds.). *Handbook of Graph Theory*. CRC Press, 2004.
11. Y. F. Hu and R. F. Blake. An improved diffusion algorithm for dynamic load balancing. *Parallel Computing*, 25(4):417–444, 1999.
12. J. G. Kemeny and J. L. Snell. *Finite Markov Chains*. Springer-Verlag, 1976.
13. F. T. Leighton. *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*. Morgan Kaufmann Publishers, 1992.
14. Stuart P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–136, 1982.
15. L. Lovász. Random walks on graphs: A survey. *Combinatorics, Paul Erdős is Eighty*, 2:1–46, 1993.
16. M. Meila and J. Shi. A random walks view of spectral segmentation. In *Eighth International Workshop on Artificial Intelligence and Statistics (AISTATS)*, 2001.
17. H. Meyerhenke, B. Monien, and S. Schamberger. Accelerating shape optimizing load balancing for parallel FEM simulations by algebraic multigrid. In *Proc. 20th IEEE Intl. Parallel and Distributed Processing Symp. (IPDPS'06)*, page 57 (CD). IEEE, 2006.
18. B. Nadler, S. Lafon, R. R. Coifman, and I. G. Kevrekidis. Diffusion maps, spectral clustering and eigenfunctions of fokker-planck operators. In *NIPS*, 2005.
19. M. Saerens, P. Dupont, F. Fouss, and L. Yen. The principal components analysis of a graph, and its relationships to spectral clustering. In *ECML 2004, European Conference on Machine Learning*, pages 371–383, 2004.
20. S. Schamberger. A shape optimizing load distribution heuristic for parallel adaptive FEM computations. In *Parallel Computing Technologies, PACT'05*, number 2763 in LNCS, pages 263–277, 2005.
21. J. R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Technical Report CMU-CS-94-125, Carnegie Mellon University, 1994.
22. The BlueGene/L Team. An overview of the BlueGene/L supercomputer. In *Proc. ACM/IEEE Conf. on Supercomputing*, pages 1–22, 2002.
23. N. Tishby and N. Slonim. Data clustering by markovian relaxation and the information bottleneck method. In *NIPS*, pages 640–646, 2000.
24. U. Trottenberg, C. W. Oosterlee, and A. Schüller. *Multigrid*. Academic Press, 2000.
25. S. van Dongen. *Graph Clustering by Flow Simulation*. PhD thesis, Univ. of Utrecht, 2000.
26. C. Xu and F. C. M. Lau. *Load Balancing in Parallel Computers*. Kluwer, 1997.
27. L. Yen, D. Vanvyve, F. Wouters, F. Fouss, M. Verleysen, and M. Saerens. Clustering using a random-walk based distance measure. In *ESANN 2005, European Symposium on Artificial Neural Networks*, 2005.