# Fast Load Balancing in Cayley Graphs

Jacques E. Boillat

Institut für Informatik und angewandte Mathematik
Universität Bern
Länggassstrasse 51
CH-3012 Bern

**Abstract.** We compare two load balancing techniques for Cayley graphs based on information and load exchange between neighboring vertices. In the first scheme, called natural diffusion, each vertex gives (or receives) a fixed part of the load difference to (from) its direct neighbors. In the second scheme, called Cayley diffusion, each vertex successively gives (or receives) a part of the load difference to (or from) direct neighbors incident to the edges labeled by the elements of the generator set of the Cayley graph. We prove that the convergence of the Cayley diffusion is faster than the natural diffusion, at least for some particular graphs (cube, circuit with an even number of vertices, graphs from the symmetric group). Furthermore, we show that the number of communications required in the Cayley diffusion is smaller than that of the natural diffusion.

**Topics covered.** Theory of Parallel and Distributed Computation, Parallel Algorithms, Load Balancing, Cayley Graphs, Complexity.

**MSC Mathematics Subject Classification.** 68R10 [Computer Science]: Discrete Mathematics in relation to Computer Science, Graph Theory. 05C25 [Combinatorics]: Graph Theory, Graphs and Groups. 05C25 [Combinatorics]: Graph Theory, Applications.

## 1 Introduction

To achieve high performance with a parallel computer it is imperative to balance the work-load of the processors. Various strategies have been proposed to solve this problem [Ha89]. We will focus here on parallel load balancing algorithms for interconnected multiprocessor networks based on local load exchange, i.e. on algorithm with the following structure:

```
Algorithm (local load balancing strategy).

  WHILE TRUE
    PAR i = 1 FOR processors
      exchange load with direct neighbors
```

There are many possibilities for local load exchange [ELZ86b], [ELZ86a], [MTS90], [Cyb89], etc. The *natural* concept is to give (or receive) a fixed part of the load difference to (from) all direct neighbors at the same time [Cyb89], [Boi90]. This

strategy should be applied if the structure of the underlying network (graph) is unknown. In [Boi90] we have shown that this method is equivalent to a time discrete Poisson equation in a finite undirected graph. We also have shown how to choose the load amount to exchange for ensuring the convergence of the algorithm in any connected undirected graph.
Each load exchange step with one particular neighbor needs 2 communications. The first for getting the neighbor's load and the second for exchanging the load. In a multiprocessor system, communication may be cost intensive and excessive synchronization steps may result in poor efficiency.
In this paper we will show that it is possible, at least for some Cayley graphs, to achieve faster convergence by exchanging load in a round robin fashion with the neighbors, and that the method requires less communication steps than the natural one.

## 2 Natural load balancing

In this section, we define the natural load balancing technique. Furthermore, we show how to choose the fixed amount of local load exchange to achieve the fastest possible convergence of the natural load balancing strategy in circuit graphs.

### 2.1 Natural load balancing

In this paper, a graph $G = (V, E)$ is a non directed, connected, and regular finite graph without loops. We will denote the adjacency matrix of $G$ by $A$ Furthermore, we will denote the order of the graph by $k$.

**Definition 1 (Natural diffusion in regular graphs).** Let $G$ be a regular graph of order $k$, and let $\alpha \in [0, 1]$ be a real number. Consider the symmetric stochastic matrix

$$P_\alpha = \alpha I + \frac{1-\alpha}{k} A \tag{1}$$

The *natural diffusion* is defined by the matrix $P_\alpha$, or more precisely by solving the equation

$$x_{t+1} = P_\alpha x_t = P_\alpha^t x_0 \quad (t \in I\!N) \tag{2}$$

where the series of powers of $P_\alpha$ denote the successive load balancing steps.

Let $j \in V$ be a vertex of $G$. We denote by $N_j$ the set of neighboring vertices of $j$. The natural diffusion corresponds to following parallel algorithm

***Algorithm 1 (Natural load balancing algorithm).***

```
WHILE TRUE
  PAR i = 1 FOR processors
    PAR j IN N_j
      give (1-α)/k times the load difference
      to direct neighbor j
```

Algorithm 1 always converges towards the uniform distribution, provided $\alpha \in ]0,1[$. It may converge if $\alpha = 0$ (see [Boi90]). Given a graph $G$ it is always possible to find $\alpha \in [0,1]$ such that the convergence of algorithm 1 is the fastest as possible. This will be shown in the next section.

*Note 2.* In practice, the load of a processor may consist in a finite sum of time complexities of (atomic) processes. If the number of atomic processes is much larger than the number of processors, we may consider the load of a processor as a real number. Tests have shown that algorithm 1 behaves well, if the processes have discrete loads, provided there are many more processes than processors [BBK91].

### 2.2 Optimal natural load balancing in circuits

In this section, we show how to chose $\alpha$ such that the convergence of the natural diffusion scheme (see algorithm 1) is optimal in a circuit. Note that the method used here is the same for any regular graph.
Let $A$ denote the adjacency matrix of the undirected circuit with $n$ vertices ($n \geq 3$), and let $\alpha \in [0,1]$ be a real number. Recall that the associated stochastic matrix is

$$P_\alpha = \alpha I + \frac{1-\alpha}{2} A \tag{3}$$

The convergence speed of this Markov chain is a function of the non stochastic eigenvalue with the greatest modulus. The eigenvalues of $P_\alpha$ are easy to compute knowing the eigenvalues of the circuit graph (see e.g. [CDS79]).

$$\lambda(P_\alpha)_j = \alpha + (1-\alpha)\cos\frac{2\pi}{n} j \qquad j = 0, ..., n-1 \tag{4}$$

If $\alpha = 1$, $P_\alpha = I$ and thus the diffusion process does not converge to the uniform load distribution. If $n$ is even and $\alpha = 0$, the smallest eigenvalue is equal to $-1$ and the diffusion process does not converge. For all other values of $\alpha$, the diffusion process will converge to the uniform load distribution.

*Question 3.* For which value of $\alpha$ is the convergence of the natural diffusion process the fastest possible?

Let $\lambda_\alpha = \max_{j=1,\dots,n} |\lambda(P_\alpha)_j|$

***Lemma 4.***

$$\lambda_\alpha = \begin{cases} \max\{|2\alpha - 1|, \alpha + (1-\alpha)\cos\frac{2\pi}{n} & n \ \textit{even} \\ \frac{1}{2}|3\alpha - 1| & n = 3 \\ \alpha + (1-\alpha)\cos\frac{2\pi}{n} & n \geq 5 \ \textit{odd} \end{cases} \tag{5}$$

*Proof.* The proof is immediate □

To answer question 3, we compute the minimal $\lambda_\alpha$ in the interval $[0,1]$

***Theorem 5.*** *Let* $\lambda = \min_{\alpha\in[0,1]} \lambda_\alpha$

$$\lambda = \begin{cases} \frac{1+\cos\frac{2\pi}{n}}{3-\cos\frac{2\pi}{n}} & n \ even \\ 0 & n = 3 \\ \cos\frac{2\pi}{n} & n \geq 5 \ odd \end{cases} \tag{6}$$

*Moreover if* $\alpha_0$ *is such that* $\lambda\alpha_0 = \lambda$, *then*

$$\alpha_0 = \begin{cases} \frac{1-\cos\frac{2\pi}{n}}{3-\cos\frac{2\pi}{n}} & n \ even \\ \frac{1}{3} & n = 3 \\ 0 & n \geq 5 \ odd \end{cases} \tag{7}$$

*Proof.* Note that

$$|2\alpha - 1| = \begin{cases} 1 - 2\alpha & \text{if } \alpha \leq \frac{1}{2} \\ 2\alpha - 1 & \text{if } \alpha \geq \frac{1}{2} \end{cases} \tag{8}$$

If $\alpha \leq \frac{1}{2}$ then it is easy to see that $1 - 2\alpha \geq \alpha + (1-\alpha)\cos\frac{2\pi}{n}$ if and only if $\alpha \leq \frac{1-\cos\frac{2\pi}{n}}{3-\cos\frac{\pi}{n}}$ If $\alpha \geq \frac{1}{2}$ then $2\alpha - 1 \geq\geq \alpha + (1-\alpha)\cos\frac{2\pi}{n}$ if and only if $\alpha = 1$. For $n$ odd, the proof is similar □

## 3 Diffusion in Cayley Graphs

**Definition 6 (Cayley Graph).** Let $\Gamma$ be a finite group and and let $S$ be a symmetric set of generators, i.e. $s \in S \Leftrightarrow s^{-1} \in S$. The Cayley Graph $G(\Gamma, S)$ is the undirected graph with vertex set $\Gamma$ and where $(g_1, g_2) \in E(G)$ if and only if $g_1^{-1}g_2 \in S$

$G(\Gamma, S)$ is a connected regular graph of order $k = |S|$. Thus at each vertex, the edges may be labeled with the elements of $S$.

*Example 1.* Let $\Gamma = (\mathbb{Z}_2)^d$, and set $S = \{(1,0,...,0),(0,1,...,0),...,(0,0,...,1)\}$. Then $G(\Gamma, S) = H_d$ the $d$-dimensional Cube Graph.

*Example 2.* Let $\Gamma = \sigma_3$ the symmetric group of the 3 element set $\{1,2,3\}$ and set $S = \{(1,2),(1,2,3),(3,2,1)\}$. Then $G(\Gamma, S)$ is a prism with triangular bottom.

*Example 3.* Consider $n \geq 2$ be an integer, the dihedral group $\Gamma = < s, t | s^2 = t^2 = (st)^n = 1 >$ of order $2n$ and set $S = \{s, t\}$. Then $G(\Gamma, S)$ is a circuit with $2n$ vertices.

We define in $S$ an equivalence relation by identifying $s$ with $s^{-1}$ ($s \in S$). Let $\hat{S}$ be the set of equivalence classes. We call $\hat{S} = \{\hat{s}_1, \hat{s}_2, ..., \hat{s}_k\}$ the set of directions of $G$. Note that if $s \in S$ is an element of order 2 then $\hat{s} = \{s\}$, if not, then $\hat{s} = \{s, s^{-1}\}$. Using $\hat{S}$, we may build a partition $\hat{E} = \{\hat{E}_1, \hat{E}_2, ..., \hat{E}_k\}$ of the edge set $E(G)$ by defining

$$(g_1, g_2) \in \hat{E}_i \text{ if and only if } g_1^{-1}g_2 \in \hat{s}_i \quad i = 1, ..., k \tag{9}$$

If $\hat{s}_i$ corresponds to an element of $S$ of order 2, then $\hat{E}_i$ is a disjoint set of vertices. If $\hat{s}_i$ corresponds to an element of $S$ of order $d > 2$, then $\hat{E}_i$ is the union of disjoint sub circuits (of length $d$) of $G$.

*Example 4.* For the $d$-dimensional Cube Graph with generator set as in example 1, $\hat{E}_i$ is the set of all edges which are parallel to the $i^{\text{th}}$ axis, i.e. $(g_1, g_2) \in \hat{E}_i$ if and only if $g_1$ and $g_2$ differ only in the $i^{\text{th}}$ component.

*Example 5.* For the symmetric group of the 3 element set $\{1,2,3\}$ with generator set as in example 2, $G(\Gamma, S)$ is a prism with triangular bottom. $\hat{E}_{(1,\hat{2})}$ corresponds to the vertical edges of $G$ and $\hat{E}_{(1,\hat{2},3)}$ corresponds to the bottom and the top of $G$.

*Example 6.* Consider the dihedral group with generator set as in example 3. If the edges of $G$ are numbered from 1 to $2n$, then $\hat{E}_s$ corresponds to the even numbered edges and $\hat{E}_t$ corresponds to the odd numbered edges.

### 3.1 Load balancing in Cayley graphs

The natural decomposition $\{\hat{E}_1, \hat{E}_2, ..., \hat{E}_k\}$ of a Cayley graph $G(\Gamma, S)$ leads to following load balancing strategy:

```
Algorithm (Cayley diffusion strategy).

    WHILE TRUE
      PAR i = 1 FOR processors
        SEQ j = 1 FOR directions
          exchange load with direct neighbors in direction j
```

Instead of distributing the load to all neighbors at the same time, the load is distributed successively in all directions. Thus a diffusion step consists in balancing the load (using algorithm 1) along all edges of the elements of the partition $\hat{E}$ successively. Two cases must be considered. If the direction $s_j$ corresponds to a generator $s$ of order 2, then each vertex gives (or receives) the half of the load difference to (from) its single direct neighbor in direction $\hat{s}_j$. If the direction $s_j$ corresponds to a generator $s$ of order $d \geq 2$, then each vertex gives (or receives) $\frac{1-\alpha_0}{2}$ times the load difference to (from) both neighbors in direction $\hat{s}_j$, where $\alpha_0$ is defined as in theorem 5. Indeed, in this case $\hat{E}_j$ consists in disjoint circuits of length $d$.
Let $\hat{G}_j$ denote the graph with the same vertex set as $G$ and with edges $\hat{E}_j$, the the algorithm may be reformulated as follows:

***Algorithm 2 (Cayley diffusion).***

```
    WHILE TRUE
      PAR i = 1 FOR processors
        SEQ j = 1 FOR directions
          exchange load using the fastest natural strategy
          with all direct neighbors in the graph Ĝj
```

Let $\hat{A}_j$ denote the adjacency matrix of the graph $\hat{G}_j$. $\hat{A}_j$ is the adjacency matrix of a disjoint union of circuits of length $d$ (if $\hat{s}_j$ corresponds to a generator of order $d > 2$) or the adjacency of a disjoint union of edges (if $\hat{s}_j$ corresponds to a generator of order $d = 2$). The stochastic matrix corresponding to the diffusion in direction $j$ is the matrix

$$\hat{P}_j = \alpha I + \frac{1-\alpha}{2}\hat{A}_j \tag{10}$$

where

$$\alpha = \begin{cases} \frac{1-\cos\frac{2\pi}{d}}{3-\cos\frac{2\pi}{d}} & d \text{ even} \\ \frac{1}{3} & d = 3 \\ 0 & d \geq 5 \text{ odd} \end{cases} \tag{11}$$

The stochastic matrix $\hat{P}$ corresponding to one step of the Cayley diffusion is the product of all partial diffusion matrices in all directions:

$$\hat{P} = \prod_{j=1}^{k} \hat{P}_j \tag{12}$$

$\hat{P}$ is a stochastic matrix, however $\hat{P}$ is normally not symmetric. Note that the eigenvalues of $\hat{P}$ are independent of the numbering of the subgraphs $\hat{G}_j$. Indeed, it is well known that given any two square matrices $A$ and $B$, $AB$ and $BA$ have the same eigenvalues. Thus the convergence of algorithm 2 is independent of the labelling of the directions.

## 4 Examples and main results

In this section, we compare both diffusion schemes for some particular Cayley graphs, e.g. the $d$-dimensional cube, the circuit with an even number of vertices and some Cayley graphs from the symmetric group. We show that the Cayley diffusion is faster than the natural diffusion in all cases.

*Example 7.* Let $G$ be square graph, i.e. the Cayley graph of $(\mathbb{Z}_2)^2$ with generators $s_1 = (1,0)$ and $s_2 = (0,1)$, then

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \tag{13}$$

$$\hat{A}_1 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \tag{14}$$

$$\hat{A}_2 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{15}$$

$$\tag{16}$$

This results in following stochastic matrices

$$P = \frac{1}{3} \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix} \tag{17}$$

$$\hat{P}_1 = \frac{1}{2} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \tag{18}$$

$$\hat{P}_2 = \frac{1}{2} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \tag{19}$$

$$\hat{P} = \frac{1}{4} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \tag{20}$$

$$\tag{21}$$

It follows that the Cayley diffusion process converges after a single step whereas the non stochastic eigenvalue of greatest modulus of the natural diffusion process is $\lambda = \frac{1}{3}$ resulting in slow convergence.

*Example 8.* The result of example 7 extends to the Cube graph of dimension $d$ (see example 1). The non stochastic eigenvalue of greatest modulus of the best possible natural diffusion process is $\lambda = 1 - \frac{2}{d+1}$ [Cyb89]. The Cayley diffusion process converges after a single step. We give here an outline of the proof. The Cayley diffusion consists of $d$ diffusion steps (successively along the edges of all directions) it follows that after $j$ steps, all vertex pairs incident to edges in direction $1, ..., j$ have the same load. In the next step, the load of both ends of these pairs may change by the same amount. The proof follows by induction.

*Example 9.* For the prism of example 2, the Cayley diffusion converges in one single step. Indeed, after the diffusion step in the direction $(1, \hat{2}, 3)$ all the vertices of the top and all vertices of the bottom have the same load. The diffusion step in the direction of $(1, \hat{2})$ equalizes the load of the top and of the bottom. It is easy to verify that the natural diffusion scheme does not converge in finite time.

*Example 10.* The circuit graph with $2n$ ($n \geq 3$) vertices may be considered as the Cayley graph of the dihedral group with generator set $S = \{s, t\}$ (see example 3). The computation of the eigenvalues of the Cayley diffusion scheme is much more harder. Let $\hat{A}_s$ and $\hat{A}_t$ be the the adjacency matrices corresponding to the subgraphs in the both dimensions, i.e.

$$\hat{A}_s = \begin{pmatrix} 0 & 1 & & & & & & \\ 1 & 0 & & & & & & \\ & & 0 & 1 & & & & \\ & & 1 & 0 & & & & \\ & & & & \cdot & \cdot & & \\ & & & & \cdot & \cdot & & \\ & & & & & & 0 & 1 \\ & & & & & & 1 & 0 \end{pmatrix} \tag{22}$$

$$\hat{A}_t = \begin{pmatrix} 0 & & & & & & & 1 \\ & 0 & 1 & & & & & \\ & 1 & 0 & & & & & \\ & & & \cdot & \cdot & & & \\ & & & \cdot & \cdot & & & \\ & & & & & 0 & 1 & \\ & & & & & 1 & 0 & \\ 1 & & & & & & & 0 \end{pmatrix} \tag{23}$$

Both $\hat{A}_s$ and $\hat{A}_t$ are involutions, i.e. $\hat{A}_s^2 = \hat{A}_t^2 = I$.

***Proposition 7.*** *Let $Q = I + \hat{A}_s + \hat{A}_t + \hat{A}_s\hat{A}_t$ then the eigenvalues of $Q$ are*

$$\lambda_j = \begin{cases} 2 + 2\cos\frac{2\pi}{n}j & j = 1, ..., n \\ 0 & j = n+1, ..., 2n \end{cases} \tag{24}$$

*Proof.* We compute the eigenvalues of $\hat{A}_s + \hat{A}_t + \hat{A}_s\hat{A}_t$. Thus we have to solve the characteristic equation $|\hat{A}_s + \hat{A}_t + \hat{A}_s\hat{A}_t - \lambda I| = 0$. Multiplying by the right with $\hat{A}_t^{-1} = \hat{A}_t$ we get $|\hat{A}_s\hat{A}_t + I + \hat{A}_s - \lambda\hat{A}_t| = 0$ and finally multiplying by the left with $\hat{A}_s^{-1} = \hat{A}_s$ we get $|\hat{A}_t + \hat{A}_s + I + \lambda\hat{A}_s\hat{A}_t| = 0$.

$$0 = \begin{vmatrix} 1 & 1 & -\lambda & & & & & & 1 \\ 1 & 1 & 1 & & & & & & -\lambda \\ & 1 & 1 & 1 & -\lambda & & & & \\ & \lambda & 1 & 1 & 1 & & & & \\ & & & \cdot & \cdot & & \cdot & \cdot & \\ & & & \cdot & \cdot & & \cdot & \cdot & \\ \lambda & & & & & & 1 & 1 & 1 \\ 1 & & & & & & -\lambda & 1 & 1 \end{vmatrix} \tag{25}$$

Adding $\lambda$ times the $2i-1^{\text{th}}$ line to the $2i^{\text{th}}$ line ($i = 1, ..., n$) we get

$$0 = \begin{vmatrix} 1 & 1 & -\lambda & & & & & 1 \\ 1+\lambda & 1+\lambda & 1-\lambda^2 & & & & & 0 \\ & 1 & 1 & 1 & -\lambda & & & \\ & 0 & 1+\lambda & 1+\lambda & 1-\lambda^2 & & & \\ & & & . & . & . & . & \\ & & & . & . & . & . & \\ \lambda & & & & & 1 & 1 & 1 \\ 1-\lambda^2 & & & & & 0 & 1+\lambda & 1+\lambda \end{vmatrix} \tag{26}$$

Thus

$$0 = (1+\lambda)^n \begin{vmatrix} 1 & 1 & -\lambda & & & & & 1 \\ 1 & 1 & 1-\lambda & & & & & \\ & 1 & 1 & 1 & -\lambda & & & \\ & & 1 & 1 & 1-\lambda & & & \\ & & & . & . & . & . & \\ & & & & . & . & . & \\ \lambda & & & & & 1 & 1 & 1 \\ 1-\lambda & & & & & & 1 & 1 \end{vmatrix} \tag{27}$$

Subtracting the $2i^{\text{th}}$ line from the $2i-1^{\text{th}}$ line $(i = 1, ..., n)$ we get

$$0 = (1+\lambda)^n \begin{vmatrix} 0 & 0 & -1 & & & & & 1 \\ 1 & 1 & 1-\lambda & & & & & \\ & 1 & 0 & 0 & -1 & & & \\ & & 1 & 1 & 1-\lambda & & & \\ & & & . & . & . & . & \\ & & & & . & . & . & \\ -1 & & & & & 1 & 0 & 0 \\ 1-\lambda & & & & & & 1 & 1 \end{vmatrix} \tag{28}$$

Adding $\lambda$ times the $2i-3^{\text{th}} \pmod{2n}$ column to the $2i+1^{\text{th}} \pmod{2n}$ column $(i = 1, ..., n)$ we get

$$0 = (1+\lambda)^n \begin{vmatrix} & & 0 & & & & & 1 \\ 1 & 1 & 1-\lambda & & 1 & & & \\ & 1 & & & 0 & & & \\ & & 1 & 1 & 1-\lambda & & 1 & \\ & & & . & & & . & \\ . & & & & . & . & . & \\ 0 & & & & & 1 & & \\ 1-\lambda & & 1 & & & & 1 & 1 \end{vmatrix} \tag{29}$$

Subtracting the $2i+1^{\text{th}}$ line from the $2i-1^{\text{th}} \pmod{2n}$ line $(i = 1, ..., n)$ we get

$$0 = (1+\lambda)^n \begin{vmatrix} & & & & & & & 1 \\ 1 & 0 & 1-\lambda & & 1 & & & \\ & 1 & & & & & & \\ & & 1 & 0 & 1-\lambda & & 1 & \\ & & & . & & & & \\ . & & & & . & . & . & \\ & & & & & 1 & & \\ 1-\lambda & & 1 & & & & 1 & 0 \end{vmatrix} \tag{30}$$

The resulting matrix has a single non zero component 1 in each $2i-1^{\text{th}}$ line and column $(i = 1, ..., n)$. Skipping these lines and columns may only change the sign of characteristic polynomial. Thus the eigenvalues remain unchanged. It remain to solve

$$0 = (1+\lambda)^n \begin{vmatrix} 1 & 1-\lambda & 1 & \\ & 1 & 1-\lambda & 1 \\ . & & . & . \\ 1-\lambda & 1 & & 1 \end{vmatrix} = (1+\lambda)^n D \tag{31}$$

The remaining determinant $D$ is the determinant of a circulant matrix. More precisely let $C_n$ be the adjacency matrix of a directed cycle with n vertices. Then $D = |I + (1-\lambda)C_n + C_n^2|$. Multiplying with $C_n^{n-1}$ the inverse of $C_n$ will not change the characteristic equation. It follows that $D = 0$ if and only if $|C_n^{n-1} + (1-\lambda)I + C_n| = 0$. It is well known (see [CDS79]) that the eigenvalues of $C_n + C_n^{n-1}$ (the undirected circuit) are $\alpha_i = 2\cos\frac{2\pi}{n}j$, $i = 1, ..., n$. Thus the solutions of $D = 0$ are $\lambda_i = 1 + 2\cos\frac{2\pi}{n}j$, $i = 1, ..., n$. Finally we get the eigenvalues of $Q$ by adding 1 to the eigenvalues of $\hat{A}_s + \hat{A}_t + \hat{A}_s\hat{A}_t$ □

***Proposition 8.*** *The Cayley diffusion process is the fastest possible, even when the symmetric process is allowed to proceed two steps.*

*Proof.* We compare the non stochastic eigenvalues of greatest modulus of the natural diffusion process (see theorem 5) and of the Cayley diffusion process. It is easy to see that $\hat{P} = (\frac{\hat{A}_s+I}{2})(\frac{\hat{A}_t+I}{2}) = \frac{1}{4}Q$. Since all eigenvalues $Q$ are positive, we have to show that

$$\left(\frac{1+\cos\frac{\pi}{n}}{3-\cos\frac{\pi}{n}}\right)^2 \geq \frac{1+\cos\frac{2\pi}{n}}{2} = \cos^2\frac{\pi}{n} \tag{32}$$

$$\frac{1+\cos\frac{\pi}{n}}{3-\cos\frac{\pi}{n}} \geq \cos\frac{\pi}{n} \tag{33}$$

$$1+\cos\frac{\pi}{n} \geq (3-\cos\frac{\pi}{n})\cos\frac{\pi}{n} \tag{34}$$

$$1 - 2\cos\frac{\pi}{n} + \cos^2\frac{\pi}{n} = \left(1-\cos\frac{\pi}{n}\right)^2 \geq 0 \tag{35}$$

□

Note that two steps of the natural diffusion process need 8 communications, and one step of the Cayley diffusion only 4 communications. Hence the Cayley diffusion is faster with only the half of communications.

*Example 11.* Let $\Gamma = \sigma_4$ the symmetric group of the 4 element set $\{1, 2, 3, 4\}$ and set $S = \{(1, 2), (1, 2, 3, 4), (4, 3, 2, 1)\}$. Then $G(\Gamma, S)$ is a cube with truncated vertices. The non stochastic eigenvalue with greatest modulus of the Cayley diffusion has value $\frac{1}{3}$ whereas that of the natural diffusion process has value 0.728553[1].

*Example 12.* Let $\Gamma = \sigma_4$ the symmetric group of the 4 element set $\{1, 2, 3, 4\}$ and set $S = \{(1, 2), (2, 3), (3, 4)\}$. Then $G(\Gamma, S)$ is a cube with truncated edges. The non stochastic eigenvalue with greatest modulus of the Cayley diffusion has value 0.577254 whereas that of the natural diffusion process has value 0.632077[1].

## 5 Conclusion

Cayley graphs make good choices for multiprocessor networks, they are vertex symmetric, and most standard networks can be formulated in terms of Cayley graphs. Moreover, we have proposed a new load diffusion scheme for Cayley graphs and have seen that its convergence is better than the natural scheme, at least for some particular graphs. In all the examples, the number of communications to achieve load balances is reduced by a factor of at least 2. We believe that this strategy is still better. In the worst case, when the generator set consists in a single element and its inverse, i.e. for cyclic groups $\Gamma = \mathbb{Z}_n$ and $S = \{1, n-1\}$, both diffusion processes are equal, thus resulting in the same convergence speed.

***Conjecture 9.*** *The convergence of the Cayley diffusion scheme is never slower than that of the natural diffusion scheme.*

We have done a lot of numerical computations with Cayley graphs from the symmetric groups. The Cayley diffusion scheme was still better than the natural one.

Another interesting area to study would be the graphs for which there exist a partition of the edges into disjoint matchings. This would lead to similar load balancing schemes.

## References


[BBK91] J.E. Boillat, F. Brugé, and P.G. Kropf. A Dynamic Load Balancing Algorithm for Molecular Dynamics Simulation on Multi–Processor Systems. *Journal of Computational Physics*, 96:1 – 14, 1991.

[Boi90] J.E. Boillat. Load Balancing and Poisson Equation in a Graph. *Concurrency: Practice and Experience*, 2(4), 1990.

[CDS79] D.M. Cvetkovič, M. Doob, and H. Sachs. *Spectra of graphs.* Academic Press, New York, 1979.


[1] This example was computed using Mathematica

[Cyb89] G. Cybenko. Dynamic Load Balancing for Distributed Memory Multiprocessors. *J. of Parallel and Distributed Computing*, 7:279–301, 1989.

[ELZ86a] D.E. Eager, E.D. Lazowska, and J. Zahorjan. A Comparison of Receiver-Initiated and Sender-Initiated Adaptive Load Sharing. *Performance Evaluation*, 6:53–68, 1986.

[ELZ86b] D.E. Eager, E.D. Lazowska, and J. Zahorjan. Adaptive Load Sharing in Homogenous Distributed Systems. *IEEE Transactions on Software Engeneering*, SE–12(5):662–675, 1986.

[Ha89] A. Hać. Load balancing in distributed systems: A summary. *Performance Evaluation Review*, 16(2-4):17–19, February 1989.

[MTS90] R. Mirchandaney, D. Towsley, and J.A. Stankovic. Adaptive Load Sharing in Heterogenous Distributed Systems. *Journal of Parallel and Distributed Computing*, 9:331–346, 1990.