VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wananga o te Upoko o te Ika a Maui

Department of Computer Science

PO Box 600
Wellington
New Zealand

Tel: +64 4 471 5328
Fax: +64 4 495 5232
Internet: Tech.Reports@comp.vuw.ac.nz

Dimension Exchange Token Distribution on the Mesh and Torus

Michael E. Houle and **Gavin Turner**

Technical Report CS-TR-96/8
May 1996

Abstract

A solution to the token distribution problem is presented for the 2-dimensional mesh and torus, based on the dimension-exchange strategy. The approach is shown to reduce the discrepancy Δ *between maximum and minimum processor loads to* δ *in optimal* $\Theta((\Delta-\delta)\cdot n)$ *time steps, where* $2 \leq \delta < \Delta$ *for the mesh, and* $4 \leq \delta < \Delta$ *for the torus.*

Publishing Information

Paper Submitted to International Symposium on Algorithms and Combinatorics (ISAAC), Osaka Japan 1996.

Author Information

Michael Houle received the Ph.D. degree in Computer Science from McGill University (Montréal, Québec, Canada) in 1989. After almost three years as a research associate at Kyushu University and the University of Tokyo in Japan, in 1992 he joined the University of Newcastle (NSW, Australia) as a lecturer in the Department of Computer Science and Software Engineering. His main research interests lie in the area of design and analysis of algorithms, in particular computational geometry, combinatorial algorithms, graph drawing, and combinatorial geometry.

Gavin Turner received the Bachelor of Computer Engineering (Hon) and the Bachelor of Computer Science (Hon) degrees at the University of Newcastle, Australia in 1991 and 1992 respectively. From 1993 to 1995 he undertook a Ph.D. program at the same institution, studying the VLSI design and analysis of load balancing systems. His Phd dissertation is currently being accessed. In 1995 he was employed as an associate lecturer in the Department of Computer Science at the University of Newcastle, Australia. He is currently a lecturer in Computer Science at Victoria University of Wellington, Wellington New Zealand. His main research interests lie in the area of design and analysis of algorithms, in particular algorithms and architectures for special-purpose parallel computing. In addition, he is interested in a wide variety of problems from image processing, load balancing and combinatorics.

Dimension-Exchange Token Distribution on the Mesh and the Torus

Michael E. Houle* Gavin Turner†

Abstract

A solution to the token distribution problem is presented for the 2-dimensional mesh and torus, based on the dimension-exchange strategy. The approach is shown to reduce the discrepancy Δ between maximum and minimum processor loads to δ in optimal $\Theta((\Delta-\delta)\cdot n)$ time steps, where $2 \leq \delta < \Delta$ for the mesh, and $4 \leq \delta < \Delta$ for the torus.

1 Introduction

One of the fundamental data distribution problems on parallel architectures is that of *token distribution*, a static variant of the well-studied load balancing problem. Each processing element (PE) of the parallel architecture possesses an initial set of *tokens*, each of which represents a task to be performed; the number of tokens stored at a particular PE is called the *load* of that PE. Ideally, one would prefer that the distribution of the tokens over the set of PEs be as even as possible, as imbalances would result in a delay in the time needed to perform all the tasks. The goal of a token distribution algorithm is to redistribute the tokens in such a way that the final loads of the PEs differ as little as possible. Here it is assumed that each token requires only a constant amount of time to send from one PE to an adjacent PE, and that no tokens are created or destroyed before the redistribution is complete.

There are many data distribution methods which achieve a balanced token distribution by gathering and making use of a certain amount of global information [3, 5, 11]. Such methods are often unsatisfactory, in that they do not take into account the practical limitations of the parallel architecture, or result in algorithms that are unnecessarily complex. One method that requires no such global information is the so-called *dimension-exchange* method, which is based on the repetitive application of an extremely simple and scalable local exchange protocol. To be able to implement a dimension-exchange algorithm on a particular parallel architecture, the communication edges of the underlying topology must be *partitionable* (or *colourable*) into sets whereby no two edges of the same set are incident on the same processor. For networks having hypercube or mesh-connected topologies, the edges can be partitioned in a natural fashion, according to the *dimension* of the network along which the edge is oriented. For other networks, partitions may be based on sets of *matchings* [4].

Dimension-exchange algorithms use the edge-colouring of a network to pair processors for data exchange, and are invariably of the following general form:

```
Dimension-Exchange Algorithm
LOOP
 FOR i = colour 1 to κ  (*κ colours *)
  Over all pairs of processors connected by edges
  of colour i, compare values and exchange;
END
```

The dimension-exchange approach has been used successfully for solutions to the problems of sorting (for example the well known algorithm of Batcher [1]), and form an integral part of many of the so-called 'hot-potato' routing algorithms [8, 10].

Due to their simplicity and scalability, many researchers have studied the applicability of dimension-exchange techniques to data distribution problems; the first being Cybenko [2] in 1987, who proposed an algorithm for the d-dimensional hypercube under the assumption that the load in each PE was *infinitely-divisible* — that is, a real-valued quantity able to be split among processors in an arbitrary fashion. Cybenko showed that if every exchange results in an equal sharing of the load be-

* *mike@cs.newcastle.edu.au*, Dept. of Computer Science, University of Newcastle, Callaghan NSW 2308, Australia

† *gavin@comp.vuw.ac.nz*, Dept. of Computer Science, Victoria University of Wellington, PO Box 600 Wellington, NZ. (Any correspondence should be sent to this address.)

tween the two PEs involved, then after d iterations the PE loads would be perfectly balanced.

This original work prompted a steady stream of research into the analysis of dimension-exchange algorithms. In 1988 Ranka, Won, and Sahni [9] studied the operation of Cybenko's algorithm empirically for the d-dimensional hypercube, under the more realistic assumption that the loads were *finitely-divisible* — that is, representable as a set of tokens. They observed that the difference between the maximum number and minimum number of tasks over all PEs of the network (called the *discrepancy*) would eventually fall to at most d. Soon after, Hosseini *et al.* [4] and Plaxton [7] confirmed this observation by providing an algorithm that after d steps reduced the discrepancy to at most d. In addition, Hosseini *et al.* demonstrated that, for infinitely-divisible loads, Cybenko's analysis could be generalized to arbitrary k-colourable networks.

In 1992 Xu and Lau [12] extended the work of Hosseini *et al.* by showing that for some topologies, the rate at which the global discrepancy converged to zero could be optimized by altering the ratio with which infinitely-divisible loads were locally balanced. Later that year the same authors [13] computed optimal ratios for the linear array, ring, 2-dimensional mesh and 2-dimensional torus.

To date, a large body of results exist detailing the performance of the dimension-exchange approach over infinitely-divisible loads; on the other hand, little has been known concerning dimension-exchange for finitely-divisible loads (tokens) on meshes and tori of constant degree. In this paper, we present asymptotically-optimal dimension-exchange algorithms for token distribution on the two-dimensional mesh and torus. For the n-by-n mesh, we prove that if the discrepancy is greater than 3, $16n$ steps of the algorithm suffice to reduce the discrepancy by 1, and if the discrepancy is equal to 3, $22n$ steps suffice. For the n-by-n torus (n even), we prove that if the discrepancy is greater than 4, $14n$ steps of the algorithm suffice to reduce the discrepancy by 1. These results are the first to establish that dimension-exchange techniques lead to optimal solutions for finitely-divisible load balancing on a mesh-connected network of constant degree.

The organization of the paper is as follows: in the next section, we describe the model of computation. In Section 3, we prove a lower bound on the complexity of the token-distribution problem, and propose dimension-exchange algorithms for the mesh and the torus. The notation and preliminary concepts that we use in the analysis of the algorithms is introduced in Section 4. The analysis of the algorithm on the torus appears in Section 5, and in Section 6, the result for the torus is extended to the mesh. Concluding remarks are made in Section 7.

2 Model of Computation

One of the simplest and most practical fixed connection networks is the one-port mesh-connected array. In this model, the processing elements (PEs) are arranged in a square grid, and are connected to their neighbours by uni-directional communication links. The PEs of the mesh may send or receive at most one message at any one time. This model is considerably weaker than the MIMD-model, where bidirectional links are assumed and concurrent communication to all the neighbours is allowed.

The interconnection network of the 2-dimensional, one-port mesh-connected array can be formally defined as follows.

Definition 1 *A 2-dimensional mesh-connected array consists of n^2 nodes arranged in an n-by-n grid. Each node represents a processor and each edge a communication link. In a toroidal configuration* (**torus**) *a node at location $[i, j]$, $0 \le i, j \le n-1$, is connected through direct communication links to 4 neighbours : $[(i\pm1) \bmod n, j]$ and $[i, (j\pm1) \bmod n]$. A* **mesh** *configuration is obtained from the torus by removing the links between $[0, j]$ and $[n-1, j]$, and $[i, 0]$ and $[i, n-1]$.*

A torus with side-length four is shown in Figure 1. In the analysis to follow, we assume that n is even.

3 The Token Distribution Problem and Algorithm

The token distribution problem $TD(A; \Delta, M, \delta)$ was first posed by Peleg and Upfal in [6] and can be stated as follows: Given parallel architecture A containing n^2 processors $P_1, ..., P_{n^2}$, with each processor P_i containing a stack of $\mu \le l(P_i) \le M$ tokens (for all $1 \le i \le n^2$) and for a global discrepancy between loads equal to $\Delta = M - \mu$, distribute the tokens such that at the end the global discrepancy has been reduced to at most δ.

3.1 A Lower Bound for the Mesh and the Torus

For a 2-dimensional mesh-connected array $\mathcal{A}$ with or without toroidal connections, a lower bound on the time required to solve token distribution problem $TD(\mathcal{A}; \Delta, M, \delta)$ is easily derived. The result is stated in Lemma 2.

Lemma 2 *Let $\mathcal{A}$ be an n-by-n mesh-connected array with or without toroidal connections. Any solution to problem* $\mathrm{TD}(\mathcal{A}; \Delta, M, \delta)$ *requires time in* $\Omega((\Delta{-}\delta) \cdot n)$.

Proof The proof is a simple bisection-width argument. Assume a worst-case load in which the $n^2/2$ PEs in one half of $\mathcal{A}$ contain M tokens, and the remaining $n^2/2$ PEs contain μ tokens. To reduce Δ to δ, $\frac{\Delta-\delta}{2} \cdot \frac{n^2}{2}$ tokens must be moved from the PEs with M tokens to the PEs with μ tokens. Given that the bisection width of $\mathcal{A}$ is $\mathcal{O}(n)$, this requires at least $\Omega((\Delta{-}\delta) \cdot n)$ steps. □

3.2 Dimension-Exchange Token Distribution Algorithm

To implement any dimension-exchange algorithm two choices must be made: the network colouring over which the algorithm will be applied, and a suitable comparison-exchange step. Many possible colourings of 2-dimensional mesh-connected arrays exist. The algorithm presented in this paper, 2DEB (2-d Dimension-Exchange Balancing), is based on the so-called *odd-even* colouring illustrated in Figure 1, in which the *odd* connections $[i, j{-}1] \leftrightarrow [i, j]$ (j odd) and $[i{-}1, j] \leftrightarrow [i, j]$ (i odd) are given colours 1 and 2, respectively, and the *even* connections $[i, j{-}1] \leftrightarrow [i, j]$ (j even) and $[i{-}1, j] \leftrightarrow [i, j]$ (i even) are given colours 3 and 4, respectively.

The local comparison-exchange step used by the algorithm can be informally stated as follows:

[Plus-Minus1]: Neighbouring PEs compare their loads and, if the loads differ, a *single token* is sent from the heavily-loaded PE to the lightly-loaded PE.

Let $\Delta = M{-}\mu$ be the initial discrepancy of the network. Algorithm 2DEB is almost exactly the same for both the mesh and the torus, the exception being that the constants appearing below are $c_0 = 16$ and $c_1 = 22$ on the mesh, and $c_0 = 14$ and $c_1 = 0$ on the torus:

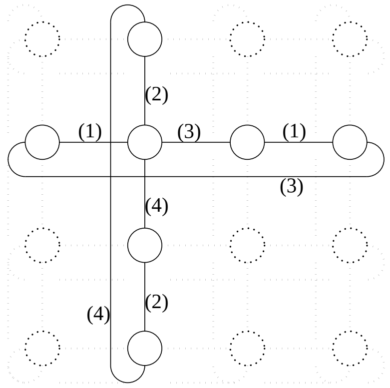

Figure 1: Odd-even colouring of the 2-dimensional torus.

Algorithm 2DEB
FOR $i = 1$ to $c_0 \cdot (\Delta{-}\delta) \cdot n + c_1 n$
  Apply the elementary step [Plus-Minus1] over all network connections having colour $(i \bmod 4){+}1$;
END

In the remainder of the paper, we shall prove two theorems that imply that 2DEB optimally decreases the discrepancy to a final value of no more than δ, where $2 \leq \delta < \Delta$ for the mesh, and $4 \leq \delta < \Delta$ for the torus.

4 Notation and Preliminaries

In this section, we present notation and observations needed for the analysis of 2DEB over the torus $\mathcal{T}$.

Consider the situation where row i of torus $\mathcal{T}$ initially contains one token per location, and all other rows contain no tokens. Over the course of $2n$ steps of 2DEB, the tokens migrate from row to row through the torus, shifting one location every other step. After the $2n$th step, the tokens are once again at their starting row i. The direction of the migration depends on the parity of i: if i is even, the row of tokens migrates from south to north; if odd, the row migrates from north to south. The columns of $\mathcal{T}$ also migrate, from east to west and from west to east.

Instead of viewing the positions of tokens as being tied to *absolute* (fixed) locations of $\mathcal{T}$, it will often be convenient to view their positions *relative* to a migrating row and a migrating column. The next subsection establishes the relationship between the absolute and the relative interpretations of Algorithm 2DEB on the torus, and the second subsection examines the behaviour of the tokens under the relative interpretation.

4.1 Row and Column Migration

Under the relative interpretation of the algorithm, the torus $\mathcal{T}$ is represented as a collection (N, S, E, W) of four sets of rows and columns: the set of north-migrating rows $N = \{N_0, N_1, \ldots, N_{\frac{n}{2}-1}\}$, the set of south-migrating rows $S = \{S_0, S_1, \ldots, S_{\frac{n}{2}-1}\}$, the set of east-migrating columns $E = \{E_0, E_1, \ldots, E_{\frac{n}{2}-1}\}$, and the set of west-migrating columns $W = \{W_0, W_1, \ldots, W_{\frac{n}{2}-1}\}$. If R is a row of $N \cup S$, and C is a column of $E \cup W$, the intersection (R, C) is associated with a *pile* of tokens, the number of which may change as each step of the algorithm is executed. The number of tokens of this pile after t steps of 2DEB will be called its *size*, and will be denoted by $\mathcal{T}(R, C)_t$. Letting $\mathcal{T}[i, j]_t$ be the number of tokens at $[i, j]$ at time t under the absolute interpretation, then initially we have $\mathcal{T}(N_i, E_j)_0 = \mathcal{T}[2i, 2j]_0$, $\mathcal{T}(S_i, E_j)_0 = \mathcal{T}[2i{+}1, 2j]_0$, $\mathcal{T}(N_i, W_j)_0 = \mathcal{T}[2i, 2j{+}1]_0$ and $\mathcal{T}(S_i, W_j)_0 = \mathcal{T}[2i{+}1, 2j{+}1]_0$, for all $0 \leq i, j < \frac{n}{2}$.

The analysis to come relies heavily upon a classification of all piles into four *directional sets* — $\mathcal{NE}$, $\mathcal{NW}$, $\mathcal{SE}$ and $\mathcal{SW}$ — that depend on the types of row and column at the intersection of which the pile is stored. $\mathcal{NE}$ is the set of all piles located at the intersection of rows of N and columns of E, and $\mathcal{NW}$, $\mathcal{SE}$ and $\mathcal{SW}$ are defined in a similar fashion. The pair of *opposite* sets $\mathcal{NE}$ and $\mathcal{SW}$ are each *orthogonal* to the opposite sets $\mathcal{NW}$ and $\mathcal{SE}$.

If a pile has size that is minimum or maximum over all positions of $\mathcal{T}$, we simply refer to it as a *minimum* or *maximum*. Let $\underline{\mathcal{NE}}_t \subset \mathcal{NE}$ and $\overline{\mathcal{NE}}_t \subset \mathcal{NE}$ be the set of minima and maxima that are members of $\mathcal{NE}$ after step t, respectively. Similarly, we define $\underline{\mathcal{NW}}_t$, $\overline{\mathcal{NW}}_t$, $\underline{\mathcal{SW}}_t$, $\overline{\mathcal{SW}}_t$, $\underline{\mathcal{SE}}_t$ and $\overline{\mathcal{SE}}_t$.

With respect to relative locations, the comparison-exchange rule of 2DEB can be reinterpreted. Consider the rows N_i of N and columns E_j of E at step $t > 0$ of 2DEB, for $0 \leq i, j < \frac{n}{2}$. If t is odd, for every row R of $N \cup S$, the piles at (R, E_j) and (R, W_k) are compared and tokens are possibly exchanged, where $k = (j{+}\frac{(t-1)}{2}) \bmod \frac{n}{2}$. If t is even, for every column C of $E \cup W$, the piles at (N_i, C) and $(S_{k'}, C)$ are compared and tokens are possibly exchanged, where $k' = (i{+}\frac{(t-2)}{2}) \bmod \frac{n}{2}$. The comparison of corresponding elements of columns E_j and W_k will be called a *swap* of E_j and W_k; similarly, we shall refer to swaps of rows. Note that in the course of n steps of 2DEB, each column E_j of E swaps with each column W_k of W exactly once, and k increases by 1 with each swap (modulo $\frac{n}{2}$); the same holds true for rows. Accordingly, we shall refer to a sequence of n consecutive steps of 2DEB as an *n-cycle.*

Whenever the piles at (R, C) and (R', C) are compared at time t, the new sizes $\mathcal{T}(R, C)_t$ and $\mathcal{T}(R', C)_t$ are recalculated as follows:

if $\mathcal{T}(R, C)_{t-1} > \mathcal{T}(R', C)_{t-1} + 1$ then
  $\mathcal{T}(R, C)_t = \mathcal{T}(R', C)_{t-1} + 1;$
  $\mathcal{T}(R', C)_t = \mathcal{T}(R, C)_{t-1} - 1;$
else if $\mathcal{T}(R, C)_{t-1} + 1 < \mathcal{T}(R', C)_{t-1}$ then
  $\mathcal{T}(R, C)_t = \mathcal{T}(R', C)_{t-1} - 1;$
  $\mathcal{T}(R', C)_t = \mathcal{T}(R, C)_{t-1} + 1;$
else
  $\mathcal{T}(R, C)_t = \mathcal{T}(R, C)_{t-1};$
  $\mathcal{T}(R', C)_t = \mathcal{T}(R', C)_{t-1};$
endif

An example of this operation is illustrated in Figure 2. A similar set of rules applies when the piles (R, C) and (R, C') are compared during a column swap.

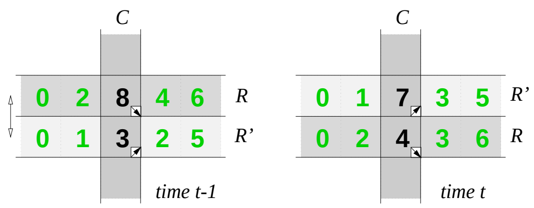

Figure 2: *Piles at (R, C) and (R', C) are compared at time t. The new sizes, $\mathcal{T}(R, C)_t$ and $\mathcal{T}(R', C)_t$ are calculated accordingly. Directional sets are indicated in the lower right-hand corner of each pile.*

The following lemma establishes the bijective relationship between absolute locations of tokens of $\mathcal{T}$ and the positions of piles relative to migrating rows and columns. The proof is straightforward, and has been omitted.

Lemma 3 *For all* $0 \leq i, j < \frac{n}{2}$,

$$\begin{aligned} \mathcal{T}(N_i, E_j)_t &= \mathcal{T}[2i{+}t', 2j{+}t'']_t, \\ \mathcal{T}(S_i, E_j)_t &= \mathcal{T}[2i{+}1{-}t', 2j{+}t'']_t, \\ \mathcal{T}(N_i, W_j)_t &= \mathcal{T}[2i{+}t', 2j{+}1{-}t'']_t \text{ and} \\ \mathcal{T}(S_i, W_j)_t &= \mathcal{T}[2i{+}1{-}t', 2j{+}1{-}t'']_t \end{aligned}$$

(indices taken modulo n), where $t' = t'' = \frac{t}{2}$ if t is even, and $t' = \frac{t-1}{2}$ and $t'' = \frac{t+1}{2}$ if t is odd.

4.2 Maxima and Minima

Consider now the situation in which $\mathcal{T}(R,C)_{t-1} = M$, the maximum possible value, for some row R and column C. If at step t R swaps with row R', we distinguish two outcomes. If $\mathcal{T}(R',C)_{t-1} \geq M-1$, then the rules outlined above indicate that after the swap, $\mathcal{T}(R,C)_t = \mathcal{T}(R,C)_{t-1} = M$ and $\mathcal{T}(R',C)_t = \mathcal{T}(R',C)_{t-1}$. Otherwise, if $\mathcal{T}(R',C)_{t-1} < M-1$, the greatest value possible for $\mathcal{T}(R,C)_t$ and $\mathcal{T}(R',C)_t$ is $M-1$. Similarly, if $\mathcal{T}(R,C)_{t-1} = \mu$, we have two cases: if $\mathcal{T}(R',C)_{t-1} \leq \mu+1$ then $\mathcal{T}(R,C)_t = \mathcal{T}(R,C)_{t-1} = \mu$ and $\mathcal{T}(R',C)_t = \mathcal{T}(R',C)_{t-1}$; if $\mathcal{T}(R',C)_{t-1} > \mu+1$ then $\mathcal{T}(R,C)_t$ and $\mathcal{T}(R',C)_t$ must be at least $\mu+1$. Noting that the same holds true for column swaps as well as row swaps, we are led to the following observation:

Observation 4 *Let M and μ be the maximum size of the piles of $\mathcal{T}$ after step t_0. For any row R and column C, if $\mathcal{T}(R,C)_{t_0} < M$, then $\mathcal{T}(R,C)_t < M$ for all $t > t_0$. Also, if $\mathcal{T}(R,C)_{t_0} > \mu$, then $\mathcal{T}(R,C)_t > \mu$ for all $t > t_0$.*

In other words, at any given position, maxima and minima can be destroyed but never created.

Consider the pile at (N_*, E_*) of $\mathcal{NE}$. In the course of an n-cycle starting after step t_0, the pile at (N_*, E_*) is compared exactly once with each of the piles at positions (S_i, E_*) and (N_*, W_j), for $0 \leq i,j < \frac{n}{2}$. If $\mathcal{T}(N_*, E_*)_{t_0} < M-1$, then the first maximum of $\overline{\mathcal{SE}}_{t_0} \cap E_*$ or $\overline{\mathcal{NW}}_{t_0} \cap N_*$ to swap with (N_*, E_*) after step t_0 will be destroyed: if such a swap does not occur by step t_0+n, the implication is that no maxima remain in $\overline{\mathcal{SE}}_{t_0+n} \cap E_*$ or $\overline{\mathcal{NW}}_{t_0+n} \cap N_*$. This follows partly from the observation that for $t > t_0$, $\mathcal{T}(N_*, E_*)_t$ cannot increase to $M-1$ without (N_*, E_*) being swapped with a maximum after step t_0. For this reason, we shall say that (N_*, E_*) *threatens* to destroy a maximum of $\overline{\mathcal{SE}}_{t_0} \cap E_*$ or $\overline{\mathcal{NW}}_{t_0} \cap N_*$. Also, if $\mathcal{T}(N_*, E_*)_{t_0} > \mu-1$, we shall say that (N_*, E_*) threatens a minimum of $\underline{\mathcal{SE}}_{t_0} \cap E_*$ or $\underline{\mathcal{NW}}_{t_0} \cap N_*$. In an analogous manner, we extend the notion of *threats* to maxima and minima to the other three directional sets.

To summarize, every pile of a directional set threatens either a minimum or a maximum (or both) of an orthogonal directional set in the same row or column. This is illustrated in Figure 3.

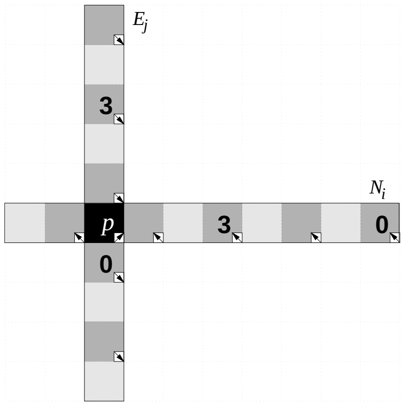

Figure 3: *Pile $p = (N_i, E_j)$ threatens the maxima of $(\overline{\mathcal{NW}}_t \cap N_i) \cup (\overline{\mathcal{SE}}_t \cap E_i)$, the minima of $(\underline{\mathcal{NW}}_t \cap N_i) \cup (\underline{\mathcal{SE}}_t \cap E_i)$, or both. These locations are darkly shaded.*

5 Analysis of 2DEB on the Torus

In this section, Algorithm 2DEB is proven to optimally solve token distribution problems $TD(\mathcal{T}; \Delta, M, \delta)$ for tori $\mathcal{T}$, and $\delta \geq 4$. Although it is easily shown that Algorithm 2DEB is unable to solve token distribution problems $TD(\mathcal{T}; \Delta, M, \delta)$ for $\delta < 4$, the proof of this claim has been omitted for the sake of brevity. The optimality of Algorithm 2DEB is stated and proven, in Theorem 10, at the end of this section. We now present several lemmas that are required to complete the proof.

Lemma 5 *Let $\mathcal{T}$ be an n-by-n torus (n even) whose elements are non-negative integers, and let M and μ be the maximum and minimum values of these elements, respectively. If $M-\mu > 2$, then after n steps of Algorithm 2DEB on $\mathcal{T}$, no row or column of $\mathcal{T}$ can contain piles α and β from different directional sets such that the size of α is μ and the size of β is M.*

Proof Without loss of generality, let $\alpha = (N_*, E_*)$ and $\mathcal{T}(N_*, E_*)_0 = \mu$, and let $\beta = (N_*, W_*)$ and $\mathcal{T}(N_*, W_*)_0 = M$. At some step $0 < t \leq n$ within the course of an n-cycle, $\mathcal{T}(N_*, E_*)_t$ is compared with $\mathcal{T}(N_*, W_*)_t$. If $\mathcal{T}(N_*, E_*)_{t-1} = \mu$ and $\mathcal{T}(N_*, W_*)_{t-1} = M$, then after the swap, $\mathcal{T}(N_*, E_*)_{t-1} = M-1$ and $\mathcal{T}(N_*, W_*)_t = \mu+1$: both the maximum and the minimum are destroyed.

Therefore either α can no longer be a minimum after step n, or β can no longer be a maximum. Since the argument holds true for any choice of α and β from different directional sets of the same row or column, the result follows. $\square$

Lemma 6 *Let $\mathcal{T}$ be an n-by-n torus (n even) whose elements are non-negative integers, and let M and μ be the maximum and minimum values of these elements, respectively. If $M-\mu > 2$, then after $9n$ steps of Algorithm 2DEB on $\mathcal{T}$, for each row and column there can be no piles α and β from the same directional set such that the size of α is μ and the size of β is M.*

Proof Let us assume that the proposition is false; that is, there exists a minimum and a maximum belonging to the same row and directional sets after $9n$ steps of the algorithm. Lemma 5 implies that this maximum and minimum must belong to different directional sets. Without loss of generality, let us assume that the maximum and minimum belong to $N_* \cap (\overline{\mathcal{NW}}_{9n} \cup \underline{\mathcal{NW}}_{9n})$, for some row N_* of N. For the purposes of the analysis, we will consider the behaviour of the algorithm through nine consecutive phases consisting of one full n-cycle each, starting at step 0.

Consider the pile (N_i, E_j), for any row N_i of N and column E_j of E, before the first step of 2DEB. The pile must threaten either maxima of $(\overline{\mathcal{NW}}_0 \cap N_i) \cup (\overline{\mathcal{SE}}_0 \cap E_i)$, minima of $(\underline{\mathcal{NW}}_0 \cap N_i) \cup (\underline{\mathcal{SE}}_0 \cap E_i)$, or both. Such a pile is illustrated in Figure 3 for $\mu = 0$ and $M = 3$. Thus in every full n-cycle, every pile of $\mathcal{NE}$ threatens maxima or minima in its row and column.

The total number τ of minima of $\underline{\mathcal{NW}}_0$ and $\underline{\mathcal{SE}}_0$, and maxima of $\overline{\mathcal{NW}}_0$ and $\overline{\mathcal{SE}}_0$ that can be destroyed by threats is at most

$$\begin{aligned} \tau &< |\mathcal{NW}| + |\mathcal{SE}| \\ &< \frac{n^2}{2}. \end{aligned}$$

Consider now the set of threats produced at positions in $\mathcal{NE}$ during the first four phases of the algorithm. A subset of these threats (call it D) will destroy maxima or minima by the end of the fifth phase. Once all of the piles of D have destroyed their maxima and minima, some of the columns of E may no longer contain maxima of $\overline{\mathcal{SE}}$ and some may no longer contain minima of $\underline{\mathcal{SE}}$. Let φ be the set of columns of E containing at least one maximum of $\overline{\mathcal{SE}}_{5n}$ and at least one minimum of $\underline{\mathcal{SE}}_{5n}$ at the end of the fifth phase. Let ε be the set of columns of E that do not contain both such maxima and minima.

For every column C of φ, all threats which exist in C during each of the four n-cycles must destroy at least one maximum or a minimum per n-cycle: there are at least $2n$ such threats over the four n-cycles. Therefore

$$2n \cdot |\varphi| \le |D| \le \tau < \frac{n^2}{2},$$

and

$$|\varphi| < \frac{n}{4} < \frac{n}{2} - |\varphi| = |\varepsilon|.$$

At the end of the fifth phase, there must be at least $\frac{n}{4}$ columns of E that contain either no minima of $\underline{\mathcal{SE}}$ or no maxima of $\overline{\mathcal{SE}}$, or both.

For phase six of the algorithm, consider any row N_* of N containing a minimum α of $\underline{\mathcal{NW}}_{9n}$, and containing a maximum β of $\overline{\mathcal{NW}}_{9n}$ as illustrated in Figure 4 for $\mu = 0$ and $M = 3$. In phase six, by the comparison-exchange rules outlined in the previous section, the comparison of minimum α with each (N_*, C_i), $C_i \in \varepsilon$, generates a pile of size at most $\mu+1$ at (N_*, C_i) (Figure 4). Since $M-\mu > 2$, this pile is a threat to the maxima of $\overline{\mathcal{NW}}_{9n}$ in N_*.

Similarly, the comparison of maximum β with each (N_*, C_i), $C_i \in \varepsilon$, generates a threat at (N_*, C_i) for the minima of $\overline{\mathcal{NW}}_{9n}$ in N_* (Figure 4). Since C_i cannot contain both maxima and minima which can be threatened by (N_*, C_i), one of the two situations will produce a threat that can only destroy a maximum or minimum of N_* — the other can be ignored.

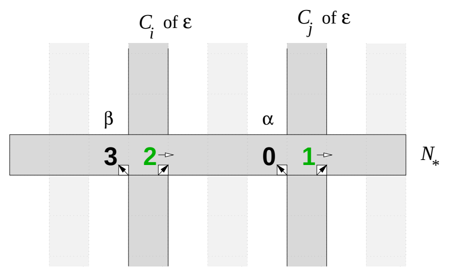

Figure 4: *Threats produced in columns C_i, $C_j \in \epsilon$ for the elements of $N_* \cap (\overline{\mathcal{NW}}_{9n} \cup \underline{\mathcal{NW}}_{9n})$.*

A total of at least $|\varepsilon|$ threats to maxima and minima of $N_* \cap (\overline{\mathcal{NW}}_{9n} \cup \underline{\mathcal{NW}}_{9n})$ are thus produced which cannot destroy maxima or minima outside N_*: one per column of $C_i \in \varepsilon$. These threats either each destroy a maximum or minimum by the end of phase seven, or exhaust either the minima or the maxima of $N_* \cap (\overline{\mathcal{NW}}_{9n} \cup \underline{\mathcal{NW}}_{9n})$. Repeating the process of phases six and seven in phases eight and nine, if there still remain both maxima and minima of $N_* \cap (\overline{\mathcal{NW}}_{9n} \cup \underline{\mathcal{NW}}_{9n})$, then at least $2 \cdot |\varepsilon| \geq \frac{n}{2}$ maxima and minima of $N_* \cap \mathcal{NW}$ must have been destroyed — a contradiction. $\square$

Lemma 7 *Let $\mathcal{T}$ be an n-by-n torus (n even) whose elements are non-negative integers, and let M and μ be the maximum and minimum values of these elements, respectively. If $M - \mu > 2$, then after $11n$ steps of Algorithm 2DEB on $\mathcal{T}$, there can be no elements α and β from opposite directional sets such that the value of α is μ and the value of β is M.*

Proof Let us assume that the proposition is false; that is, there exists a minimum and a maximum belonging to opposite directional sets after $11n$ steps of the algorithm. Without loss of generality, let us assume that the minimum belongs to $N_* \cap \underline{\mathcal{NE}}_{11n}$, and the maximum belongs to $S_* \cap \overline{\mathcal{SW}}_{11n}$ for some rows N_* of N and S_* of S.

Row N_* will swap with S_* at some step t, $9n < t \leq 10n$. This event is illustrated in Figure 5 for $M = 3$ and $\mu = 0$. Let I and J be index sets such that for all $i \in I$ and $j \in J$, N_* contains an element of $\underline{\mathcal{NE}}_t$ at (N_*, E_i), and S_* contains an element of $\overline{\mathcal{SW}}_t$ at (S_*, W_j).

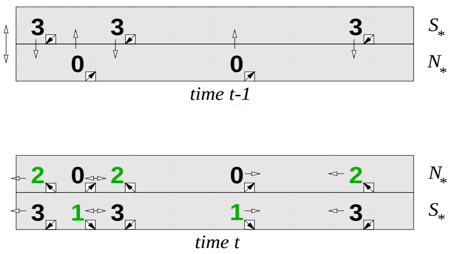

Figure 5: *Rows N_* and S_* swap at time t. Maxima and minima are shown in black and threats shown in grey. Directional sets are indicated in the corner of each cell.*

Any threat to minima in row N_* must belong to $\mathcal{NW}$ and have size at least $\mu+2$, and any threat to maxima in row S_* must belong to $\mathcal{SE}$ and have size at most $M-2$.

After the row swap at step t, the pile at (S_*, E_i) has size at most $\mu+1$, and since $M-\mu > 2$, it must threaten maxima in $S_* \cap \overline{\mathcal{SW}}_{9n}$ and $E_j \cap \overline{\mathcal{NE}}_{9n}$; however, since Lemmas 5 and 6 imply that there exists no maximum in E_i after step $9n$, the threat can only destroy a maximum of $S_* \cap \overline{\mathcal{SW}}_{9n}$. Similarly, a threat is created at (N_*, W_j) that can destroy only minima of $N_* \cap \underline{\mathcal{NE}}_{9n}$. The number of such threats created for minima in $N_* \cap \underline{\mathcal{NE}}_{9n}$ is $|S_* \cap \overline{\mathcal{SW}}_{9n}|$, and the number created for maxima in $S_* \cap \overline{\mathcal{SW}}_{9n}$ is $|N_* \cap \underline{\mathcal{NE}}_{9n}|$. As a result, either all minima from row N_* or all maxima from row S_* (whichever is the lesser) must be removed from the distribution by step $t+n$. $\square$

Lemma 8 *Let $\mathcal{T}$ be an n-by-n torus (n even) whose elements are non-negative integers, and let M and μ be the maximum and minimum values of these elements, respectively. If $M - \mu > 4$, then after $14n$ steps of Algorithm 2DEB on $\mathcal{T}$, there can be no elements α and β from the same directional set such that the value of α is μ and the value of β is M.*

Proof Let us assume that the proposition is false; that is, there exists a minimum and a maximum belonging to the same directional sets after $14n$ steps of the algorithm. After step $9n$ Lemma 6 gaurantees that these maxima and minima cannot share the same row or column. Without loss of generality, let us assume that the maximum belongs to $N_* \cap \overline{\mathcal{NE}}_{14n}$, and the minimum belongs to $E_* \cap \underline{\mathcal{NE}}_{14n}$ for some row N_* of N and column E_* of E.

After step $11n$ Lemma 7 guarantees that no minima or maxima belong to $\mathcal{SW}$. We consider the behaviour of the algorithm through two consecutive n-cycles beginning after step $11n$.

A threat for maxima of $N_* \cap \overline{\mathcal{NE}}_{14n}$ is one that belongs to $\mathcal{NW}$ having height at least $\mu+2$, and a threat for minima of $E_* \cap \underline{\mathcal{NE}}_{14n}$ is one that belongs to $\mathcal{SE}$ having height at most $M-2$.

Consider the action of the algorithm at step t, for any $11n < t < 13n$ and t odd. At step t, column E_* is swapped with some column W_* from W, and at step $t+1$, row N_* is swapped with some row S_* from S. If $\mathcal{T}(N_*, E_*)_{t-1} < M-2$, then $\mathcal{T}(N_*, W_*)_t \leq M-2$, and the pile at (N_*, W_*) thereby becomes a threat to maxima in $N_* \cap \mathcal{NE}$ — since there are no maxima in $\overline{\mathcal{SW}}_{11n}$, the pile cannot destroy any maxima in W_*. This situation is illustrated in Figure 6(a) for $\mu = 0$ and $M = 5$. The same thing happens if $\mathcal{T}(N_*, E_*)_{t-1} = M-2$ and $\mathcal{T}(N_*, W_*)_{t-1} \leq M-2$.

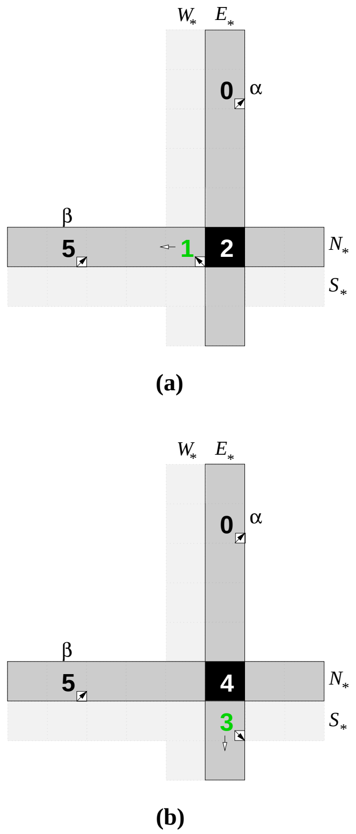

Figure 6: *A threat is generated by (N_*, E_*) in either N_* or E_* after every two steps of 2DEB.*

On the other hand, if $\mathcal{T}(N_*, E_*)_{t-1} > M-2$, or if $\mathcal{T}(N_*, E_*)_{t-1} = M-2$ and $\mathcal{T}(N_*, W_*)_{t-1} > M-2$, then $\mathcal{T}(N_*, E_*)_t \geq M-2$. Since $M-\mu > 4$, this implies that $\mathcal{T}(N_*, E_*)_t > \mu+2$. If so, then $\mathcal{T}(S_*, E_*)_t \geq \mu+2$, and the pile at (S_*, E_*) thereby becomes a threat to minima in $E_* \cap \mathcal{NE}$ — and since there are no minima in $\underline{\mathcal{SW}}_{11n}$, the pile cannot destroy any minima in S_*. This situation is illustrated in Figure 6(b).

As a result of these two steps, either a threat for the minima in column E_* is created at (S_*, E_*), or a threat for the maxima in row N_* is created in (N_*, W_*). Since n threats are created in N_* and E_* ($\frac{n}{2}$ per n-cycle), and N_* and E_* can each absorb at most $\frac{n}{2}$ threats, then after $14n$ steps of 2DEB, either no maxima of N_* remain, or no minima of E_* remain. We therefore have a contradiction. $\square$

Lemma 9 *Let $\mathcal{T}$ be an n-by-n torus (n even) whose elements are non-negative integers, and let M and μ be the maximum and minimum values of these elements, respectively. If $M-\mu > 3$, then after $8n$ steps of Algorithm 2DEB on $\mathcal{T}$, there can be no elements α and β from orthogonal directional sets such that the value of α is μ and the value of β is M.*

Proof Let us assume that the proposition is false; that is, there exists a minimum and a maximum belonging to orthogonal directional sets after $8n$ steps of the algorithm. Without loss of generality, let us assume that the maximum belongs to $\overline{\mathcal{SE}}_{8n}$, and the minimum belongs to $\underline{\mathcal{NE}}_{8n}$. For the purposes of the analysis, we will consider the behaviour of the algorithm through eight consecutive phases consisting of one full n-cycle each.

After the first phase, Lemma 5 guarantees that no row or column contains both maxima and minima from different directional sets. Let $\underline{\mathcal{NE}}^c$, $\underline{\mathcal{SW}}^c$, $\overline{\mathcal{SE}}^c$ and $\overline{\mathcal{NW}}^c$ be the sets of columns containing elements of $\underline{\mathcal{NE}}_n$, $\underline{\mathcal{SW}}_n$, $\overline{\mathcal{SE}}_n$ and $\overline{\mathcal{NW}}_n$, respectively. Similarly, we define $\underline{\mathcal{NE}}^r$, $\underline{\mathcal{SW}}^r$, $\overline{\mathcal{SE}}^r$ and $\overline{\mathcal{NW}}^r$ as the sets of rows containing elements of $\underline{\mathcal{NE}}_n$, $\underline{\mathcal{SW}}_n$, $\overline{\mathcal{SE}}_n$ and $\overline{\mathcal{NW}}_n$, respectively. Each row and column appears in at most one of the above sets.

During any given full n-cycle, each row N_i of N swaps with every row S_j of S exactly once. As a result of the swap, in every column W_k of W, either a threat to minima of $\underline{\mathcal{NE}}_n \cap N_i$ and $\underline{\mathcal{SW}}_n \cap W_k$ exists at (N_i, W_k), or a threat to maxima of $\overline{\mathcal{SE}}_n \cap S_j$ and $\overline{\mathcal{NW}}_n \cap W_k$ exists at (S_j, W_k), or both.

To estimate the number of distinct threats produced in W_k during a full n-cycle, consider the following counting argument. We maintain a score for each row of W_k; initially these scores are all zero. At every swap involving row N_i, if (N_i, W_k) stores a threat to minima, we increase its score by one; the same is done for rows of S. The total of the scores is at least as large as the number of swaps of rows of N with rows of S: $\frac{n^2}{4}$. Since each row is involved in exactly $\frac{n}{2}$ swaps, a particular row can increase its score at most $\frac{n}{2}$ times. The total number of distinct threats produced in column W_k throughout a full n-cycle is therefore at least $(\frac{n^2}{4})/(\frac{n}{2}) = \frac{n}{2}$. Note that no row of N or S can absorb more than one threat from any given column of W.

The total number τ of minima of $\underline{\mathcal{NE}}_n$ and $\underline{\mathcal{SW}}_n$, and maxima of $\overline{\mathcal{NW}}_n$ and $\overline{\mathcal{SE}}_n$ that can be destroyed by these threats is at most

$$\begin{aligned}\tau &\leq |\underline{\mathcal{NE}}^r| \cdot |\underline{\mathcal{NE}}^c| + |\underline{\mathcal{SW}}^r| \cdot |\underline{\mathcal{SW}}^c| + \\ &\quad |\overline{\mathcal{NW}}^r| \cdot |\overline{\mathcal{NW}}^c| + |\overline{\mathcal{SE}}^r| \cdot |\overline{\mathcal{SE}}^c| \\ &\leq \frac{n}{2} \cdot |\underline{\mathcal{NE}}^c| + \frac{n}{2} \cdot |\underline{\mathcal{SW}}^c| + \\ &\quad \frac{n}{2} \cdot |\overline{\mathcal{NW}}^c| + \frac{n}{2} \cdot |\overline{\mathcal{SE}}^c|\end{aligned}$$

$$\leq \quad \frac{n^2}{2}.$$

Consider now the set of threats produced in the columns of W, and the rows of N and S, during the second through fifth phases of the algorithm. A subset of these threats (call it D) will destroy a maximum or minimum by the end of the sixth phase, the others will not. Once all of the piles of D have destroyed their maximum or minimum, some of the columns of W may no longer contain maxima or minima of $\overline{\mathcal{NW}}$ or $\underline{\mathcal{SW}}$. Let φ be the set of columns of W containing at least one maximum of $\overline{\mathcal{NW}}_{6n}$ or minimum of $\underline{\mathcal{SW}}_{6n}$ at the end of the sixth phase. Let ε be the set of columns of W that do not contain such maxima or minima.

For every column C of φ, all threats produced in C over the four n-cycles must destroy a maximum or a minimum: there are at least $2n$ such threats. Therefore

$$2n \cdot |\varphi| \leq |D| < \tau \leq \frac{n^2}{2},$$

and

$$|\varphi| < \frac{n}{4} < \frac{n}{2} - |\varphi| = |\varepsilon|.$$

At the end of the sixth phase, there must be at least $\frac{n}{4}$ columns of W that contain no minima of $\underline{\mathcal{SW}}$ or maxima of $\overline{\mathcal{NW}}$.

For phases seven and eight of the algorithm, consider any row N_* of N containing a minimum of $\underline{\mathcal{NE}}_{8n}$, and any row S_* of S containing a maximum of $\overline{\mathcal{SE}}_{8n}$. In phase seven, the columns of ε generate a total of at least $\frac{n}{4}$ threats that can only destroy maxima of $\overline{\mathcal{SE}}$ in S_* and minima of $\underline{\mathcal{NE}}$ in N_*. Let ν and σ be the number of threats produced by ε in N_* and S_*, respectively. Since some maximum of $S_* \cap \overline{\mathcal{SE}}_{8n}$ and some minimum of $N_* \cap \underline{\mathcal{NE}}_{8n}$ must survive, we have

$$\frac{n}{4} < \nu + \sigma < |S_* \cap \overline{\mathcal{SE}}_{6n}| + |N_* \cap \underline{\mathcal{NE}}_{6n}| \leq |\varphi| < \frac{n}{4},$$

which is a contradiction. □

Theorem 10 *Let $\mathcal{T}$ be an n-by-n torus whose elements are non-negative integers, and let M and μ be the maximum and minimum values of these elements, respectively. If $M-\mu > 4$, then after $14n$ steps of Algorithm 2DEB on $\mathcal{T}$, there can be no elements α and β such that the value of α is μ and the value of β is M.*

Proof The proof follows immediately from Lemmas 3, 7, 8 and 9 which together guarantee that after step $14n$ there cannot exist such an α or a β in identical, in opposite, or in orthogonal directional sets. □

Corollary 11 *Algorithm 2DEB reduces the discrepancy of the n-by-n torus $\mathcal{T}$ from $\Delta = M-\mu$ to $\delta \geq 4$ in $\mathcal{O}((\Delta-\delta) \cdot n)$ time.*

6 Extension of Analysis to the Mesh

In this section, we show how the results of Section 5 for the torus lead directly to prove that Algorithm 2DEB optimally solves token distribution problems $TD(\mathcal{M}; \Delta, M, \delta)$ for mesh $\mathcal{M}$, and $\Delta \geq \delta \geq 2$. The result is obtained via a simulation of the mesh by a torus of twice the sidelength, upon which the results of Section 5 are applied. It can easily be shown that Algorithm 2DEB is unable to solve token distribution problems $TD(\mathcal{T}; \Delta, M, \delta)$ for $\delta < 2$; however, the proof of this claim has been omitted for the sake of brevity.

Unlike the analysis of the previous section for the torus, the analysis for the mesh uses the original, absolute interpretation of the algorithm. Let $\mathcal{M}[a,b]_t$ be the value stored at $[a,b]$ of $\mathcal{M}$ after t steps of Algorithm 2DEB on $\mathcal{M}$, and let $\mathcal{T}[a,b]_t$ be the value stored at $[a,b]$ of $\mathcal{T}$ after t steps of Algorithm 2DEB on $\mathcal{T}$.

Given an n-by-n mesh $\mathcal{M}$, we construct a $2n$-by-$2n$ torus $\mathcal{T}$ and initialize it as follows: for all $0 \leq i, j \leq n-1$, we set

$$\begin{aligned} &\mathcal{T}[i,j]_0 = \mathcal{T}[2n-1-i,j]_0 = \mathcal{T}[i,2n-1-j]_0 \\ &= \mathcal{T}[2n-1-i,2n-1-j]_0 = \mathcal{M}[i,j]_0. \end{aligned}$$

The southeast quadrant of $\mathcal{T}$ contains a copy of $\mathcal{M}$, and the remaining quadrants contain mirror images of $\mathcal{M}$ reflected about imaginary lines between the $(n-1)$st and nth rows, and the $(n-1)$st and nth columns.

Lemma 12
For all $0 \leq i, j \leq n-1$, $\mathcal{T}[i,j]_t = \mathcal{T}[2n-1-i,j]_t = \mathcal{T}[i,2n-1-j]_t = \mathcal{T}[2n-1-i,2n-1-j]_t = \mathcal{M}[i,j]_t$.

Proof By induction on t. By definition, the lemma holds when $t = 0$.

Assume that the lemma holds for $t = t'-1 \geq 0$. We shall show that the lemma is true for $t = t'$.

Without loss of generality, assume that step t of Algorithm 2DEB is a vertical exchange. There are two basic cases:

- *Row i of $\mathcal{M}$ participates in an exchange at step t'.*

  Let us assume that row i exchanges tokens with row $i{+}1$, in both $\mathcal{M}$ and $\mathcal{T}$. The induction hypothesis implies that

$$\begin{aligned}\mathcal{M}[i,j]_{t'-1} &= \mathcal{T}[i,j]_{t'-1}\\ &= \mathcal{T}[2n-1-i,j]_{t'-1}\\ &= \mathcal{T}[i,2n-1-j]_{t'-1}\\ &= \mathcal{T}[2n-1-i,2n-1-j]_{t'-1}\end{aligned}$$

  and

$$\begin{aligned}\mathcal{M}[i{+}1,j]_{t'-1} &= \mathcal{T}[i{+}1,j]_{t'-1}\\ &= \mathcal{T}[2n-2-i,j]_{t'-1}\\ &= \mathcal{T}[i{+}1,2n-1-j]_{t'-1}\\ &= \mathcal{T}[2n-2-i,2n-1-j]_{t'-1}.\end{aligned}$$

  Since $2n-2-2i$ is even, row $2n-2-i$ of $\mathcal{T}$ exchanges tokens with row $2n-1-i$. Consequently,

$$\begin{aligned}\mathcal{M}[i,j]_{t'} &= \mathcal{T}[i,j]_{t'}\\ &= \mathcal{T}[2n-1-i,j]_{t'}\\ &= \mathcal{T}[i,2n-1-j]_{t'}\\ &= \mathcal{T}[2n-1-i,2n-1-j]_{t'}\end{aligned}$$

  and

$$\begin{aligned}\mathcal{M}[i{+}1,j]_{t'} &= \mathcal{T}[i{+}1,j]_{t'}\\ &= \mathcal{T}[2n-2-i,j]_{t'}\\ &= \mathcal{T}[i{+}1,2n-1-j]_{t'}\\ &= \mathcal{T}[2n-2-i,2n-1-j]_{t'}.\end{aligned}$$

- *Row i of $\mathcal{M}$ does not participate in an exchange at step t'.*

  Here, the row is on the border of the mesh; that is, either $i=0$ or $i=n-1$. Let us assume that $i=n-1$; the case where $i=0$ is handled analogously.

  In the mesh, the values of row $n-1$ do not change from step $t'-1$ to step t'. In the torus, an exchange occurs between rows $n-1$ and n. Since the induction hypothesis implies that for all $0 \le j \le n-1$, $\mathcal{M}[n-1,j]_{t'-1} = \mathcal{T}[n-1,j]_{t'-1} = \mathcal{T}[n,j]_{t'-1} = \mathcal{T}[n-1,2n-1-j]_{t'-1} = \mathcal{T}[n,2n-1-j]_{t'-1}$, the values of rows $n-1$ and n of $\mathcal{T}$ also do not change from step $t'-1$ to step t'. $\square$

Theorem 13 *Let $\mathcal{M}$ be an n-by-n mesh whose elements are non-negative integers, and let M and μ be the maximum and minimum values of these elements, respectively. If $M-\mu > 3$, then after $16n$ steps of Algorithm 2DEB on $\mathcal{M}$, there can be no elements α and β such that the value of α is μ and the value of β is M. If $M-\mu = 3$, then $22n$ steps suffice.*

Proof Assume that such elements α and β do exist, and that $\alpha = \mathcal{M}[i,j]$ for some $0 \le i,j \le n-1$. Let $\mathcal{T}$ be the $2n$-by-$2n$ torus constructed from $\mathcal{M}$ as described above. The maximum and minimum values of elements of $\mathcal{T}$ are also M and μ.

For any integers n and a, the difference between a and $2n-1-a$ is an odd number, $2n-1-2a$. This implies that the values stored at $\mathcal{T}[i,j]$, $\mathcal{T}[2n-1-i,j]$, $\mathcal{T}[i,2n-1-j]$ and $\mathcal{T}[2n-1-i,2n-1-j]$ all belong to different directional sets, and the previous lemma implies that the values they store at any time are all equal to μ.

One of these four directional sets must be orthogonal to that of an element storing β, and another two must be opposite. Applying Lemma 9 when $M-\mu > 3$, and Lemma 7 when $M-\mu = 3$, we arrive at a contradiction. $\square$

Corollary 14 *Algorithm 2DEB reduces the discrepancy of the n-by-n mesh $\mathcal{M}$ from $\Delta = M-\mu$ to $\delta \ge 2$ in $\mathcal{O}((\Delta-\delta)\cdot n)$ time.*

7 Conclusion

In this paper, we presented a dimension-exchange data distribution algorithm and proved that it is asymptotically-optimal for token distribution on the two-dimensional mesh and torus. The benefits of the dimension-exchange approach, in that it is extremely simple, uses only locally-available information and is completely scalable, cannot be overstated. The analysis shows for the first time that dimension-exchange techniques can lead to optimal solutions for token distribution on mesh-connected networks of constant degree.

This paper is also the first to propose and make use of a novel comparison-exchange step in which a single token (task) is transferred between neighbouring processors. In all previous papers, [2, 4, 7], a greedy approach has been taken, in which each local step attempts to equalize the local loads of neighbouring PEs. The merit of our approach is twofold. Firstly, it provides an algorithm that is capable of solving token distribution problems $TD(\mathcal{A}; \Delta, M, \delta)$ for all possible $\delta \geq \gamma$ (where γ represents the minimum degree of the network). All previous solutions have been capable of solving only problem $TD(\mathcal{A}; \Delta, M, \gamma)$. Secondly, it greatly simplifies the implementation of the algorithm. After each step of a greedy approach, synchronization between the processors is required since an unequal, indeterminable number of tokens may be exchanged by different pairs of PEs. Some form of global checking would be required (either in hardware or software) to determine when all comparison-exchange steps have been completed. 2DEB, requiring no such checking, can be implemented on the simplest of parallel hardware and is therefore well-suited to massively-parallel computing applications.

Despite the significant difference between 2DEB and previous approaches, one interesting similarity can be observed. In each case analysis reveals that, under the operation of the dimension-exchange process, the discrepancy between maximum and minimum loads converges to a limit equivalent to the minimum degree of the underlying network. An interesting direction for future research would be to investigate the link between the degree of a network and this limit of convergence.

Simulation of Algorithm 2DEB suggests that, even in the worst case, the time required to reduce the discrepancy by 1 on the torus and mesh is much closer to their respective bisection lower bounds. Another possibility for future research in this area lies in refining the analysis, with the aim of reducing the constants of proportionality of the algorithm.

References

[1] K. Batcher. Sorting networks and their applications. In *Proceedings AFIPS Spring Joint Conference*, volume 32, pages 307–314, 1968.

[2] G. Cybenko. Dynamic load balancing for distributed memory multiprocessors. *Journal of Parallel and Distributed Computing*, 7:279–301, 1989.

[3] D. Diderich and H. Gengler. An efficient algorithm for solving the token distribution problem on k-ary d-cube networks. Technical Report LITH-97, Swiss Federal Institute of Technology, Lausanne, Switzerland, March 1994.

[4] S.H. Hosseini, B. Litow, M. Malkawi, J. McPherson, and K. Vairavan. Analysis of graph coloring based distributed load balancing algorithm. *Journal of Parallel and Distributed Computing*, 10:160–166, 1990.

[5] F. Meyer auf der Heide, B. Oesterdiekhoff, and R. Wanka. Strongly adaptive token distribution. In *Proceedings of the 20th ICALP*, pages 398–409, 1993.

[6] D. Peleg and E. Upfal. The generalised packet routing problem. *Theoretical Computer Science*, 53:218–293, 1987.

[7] C. G. Plaxton. Load balancing, selection and sorting on the hypercube. In *Proceedings of the ACM-SPAA*, pages 64–73, 1989.

[8] R. T. Plunkett and A. Symvonis. On the hot-potato permutation routing algorithm of borodin, rabani and schieber. In *Proceedings of Computing: The Australasian Theory Symposium (CATS '96)*, Jan 1996.

[9] S. Ranka, Y. Won, and S. Sahni. Programming a hypercube multicomputer. *IEEE Software*, pages 69–77, 1988.

[10] A. Roberts and A. Symvonis. On deflection worm routing on meshes. In *Proceedings of the First IEEE International Conference on Algorithms and Architecture for Parallel Processing (ICA3PP)*, pages 375–378, Apr 1995.

[11] G. Turner and H. Schröder. Token distribution on reconfigurable d-dimensional meshes. In *Proceedings of the 1st IEEE International Conference on Algorithms and Architectures for Parallel Processing*, volume 1, pages 335–344, 1995.

[12] C.Z. Xu and F.C.M. Lau. Analysis of the generalized dimension exchange method for dynamic load balancing. *Journal of Parallel and Distributed Computing*, 16:385–393, 1992.

[13] C.Z. Xu and F.C.M. Lau. The generalized dimension exchange method on some specific structures. Technical Report TR-92-02, University of Hong Kong, 1992.