Dimension-Exchange Algorithms for Load Balancing on Trees*

MICHAEL E. HOULE
IBM Research, Japan
The University of Sydney, Australia

ANTONIOS SYMVONIS
University of Ioannina, Greece

DAVID R. WOOD
Carleton University, Canada

Abstract

This paper considers dimension-exchange algorithms for load balancing on trees with finitely-divisible loads (*token distribution*). We present improved analysis of an existing protocol, and in particular, establish a logarithmic upper bound on the discrepancy of the final distribution. Our second contribution is a new algorithm, which assuming each node has knowledge of the total number of nodes, determines a perfectly balanced distribution.

Keywords

load balancing, token distribution, dimension-exchange, tree

1 Introduction

A fundamental data distribution problem on networks of processors is that of *load balancing*. Each processor possesses an initial *load*, which represents an amount of work to be performed. To minimise the time needed to perform all tasks, one desires that the load be evenly distributed over all processors. Thus, the goal of a load balancing algorithm is to redistribute the load in such a way that the maximum difference between the loads of two processors (the *discrepancy*) is minimised. In this paper we consider *static* load balancing, in which it is assumed that the total load is fixed, and no load is created or destroyed before the redistribution is complete. Furthermore we assume *synchronous* communication, in which each processor runs a clock and all these clocks are in step. We model the

*Research completed at the School of Information Technologies, The University of Sydney, Australia, and supported by the Australian Research Council Large Grant A49906214.

communication network by a simple undirected graph, whose nodes correspond to processors and edges correspond to communication links.

Types of Load Balancing Algorithms: Load balancing schemes can be distinguished according to the following four criteria. The first criterion discriminates between global and local computation. Some data distribution methods employ a centralised processor to gather and make use of a certain amount of *global* information. Such methods are often unsatisfactory, in that they do not take into account the practical limitations of network communication, or result in unnecessarily complex algorithms. In a *local* load balancing scheme, each processor is running the same protocol which can only make use of locally-available information, such as the load at that processor and at its neighbours. The second criterion concerns the number of neighbouring processors with which a given processor can communicate at each step. In a *single-port* model each processor may send and receive at most one message at any one time. This model is considerably weaker than the *multi-port* model, also called the Multiple Instruction Multiple Data (MIMD) model, where concurrent communication to all the neighbours is allowed. The third criterion deals with how load itself is modelled. One alternative is to assume that the load is *infinitely-divisible*; that is, a real-valued quantity able to be arbitrarily split among processors. A more realistic model considers the load to consist of unit-sized jobs (called *tokens*). In this model, load balancing is also called *token distribution*. The final criterion, which is only applicable in a token distribution setting, deals with the number of tokens which can be transfered long a communication link in a single step. In the *multiple-token* model an arbitrary number of tokens can be transfered, while in the *single-token* model at most one token can be transfered across an edge in a single step.

In this paper, we are consider token distribution in a single-port single-token model mainly using local computation only. (In Section 4 we describe a token distribution algorithm which uses some global information to perform optimal balancing.) One method for token distribution in the single-port model that requires no global information is the so-called *dimension-exchange* method. Here the edges of the network are coloured in a preprocessing step such that no two edges incident to a common node receive the same colour. (The classical result of Vizing [11] states that a simple graph with maximum degree D has such an edge-colouring with D or $D+1$ colours.) The copy of the algorithm running at node v uses the colouring of edges incident to v in order to pair processors for data exchange. Dimension-exchange algorithms are invariably of the following general form, where the set of edge colours is taken to be $\{0,1,\ldots,\chi-1\}$.

Algorithm DIMENSION-EXCHANGE (node v)
  $t \leftarrow 0$;
  repeat
    if there exists an edge vw of colour $t \bmod \chi$ incident to v **then**
      exchange information on loads between v and w;
      compare the loads of v and w according to some protocol;
      if required, send token(s) from v to w or receive token(s) from w;
    end-if
    $t \leftarrow t+1$;
  until some stopping condition is satisfied;

For the dimension-exchange protocols described in this paper, the body of the 'if' statement in the DIMENSION-EXCHANGE algorithm can be implemented in parallel across all nodes in a constant number of communication steps. We therefore consider these steps to be executed in one unit of time. In each step, those edges of the colour under consideration are said to be *active*. A sequence of χ consecutive steps is called a *round*. (During a round every edge is active exactly once.)

Our Results: In this paper we consider token-distribution on trees. This approach is of wider significance since an algorithm for token-distribution on trees can be applied to a spanning tree of an arbitrary network. The first contributions of this paper is improved analysis of an existing dimension-exchange protocol for arbitrary trees. Previous analysis of this protocol on trees has been for the complete binary tree only. For a given tree T, we determine the worst case distribution on T under this protocol. We then prove that for an arbitrary initial distribution on an n-node tree T with maximum degree D, this protocol will reduce the discrepancy to at most

$$\min\left\{ \left\lfloor \frac{n}{2} \right\rfloor, 1+(D-2)\lceil \log_2 n \rceil, \left\lfloor \frac{D+1}{2} \lceil \log_2 n \rceil \right\rfloor \right\} .$$

As an example, we show that this protocol will reduce the discrepancy of a distribution on the complete k-ary tree of height h ($k \geq 1$, $h \geq 1$) to at most $\min\{(k+1)(h+1)/2, (k-1)h+1\}$. This generalises the result of Houle, Tempero, and Turner [5].

Our second contribution is a new dimension-exchange algorithm which produces a distribution with discrepancy at most one, which is of course optimal. For trees of bounded degree, the rate of convergence is shown to be optimal in the worst-case. Unfortunately, this algorithm assumes that each node has knowledge of the number of nodes in the tree. This is the first known algorithm for single-token single-port load balancing which achieves optimal discrepancy.

Related Work: A dimension-exchange algorithm for load balancing on the d-dimensional hypercube with infinitely divisible loads was proposed by Cybenko [1]. Cybenko showed that if every exchange results in an equal sharing of the load between the two nodes involved, then after d iterations the discrepancy would be zero. Hosseini *et al.* [4] demonstrated that, for infinitely-divisible loads, Cybenko's analysis could be generalised to arbitrary networks.

Assuming finitely-divisible loads, Hosseini *et al.* [4] provided a dimension-exchange algorithm for token distribution on the d-dimensional hypercube which, after d steps, reduced the discrepancy to at most d. Houle and Turner [6] proposed a dimension-exchange algorithm for the two-dimensional mesh and torus, which reduces the discrepancy to two for the mesh and four for the torus, both in worst-case optimal time. The same algorithm is analysed by Houle, Tempero, and Turner [5] for token distribution on the complete binary tree. They showed that the discrepancy converges to at most the height of the tree, again in optimal time in the worst case.

Ghosh and Muthukrishnan [3] and Ghosh *et al.* [2] studied a randomised dimension-exchange algorithm for token distribution on arbitrary graphs (as well as a deterministic multi-port algorithm). Their algorithm determines a random matching at each step, as opposed to cycling through the edges with respect to a fixed edge-colouring. Note that the result in [2] only guarantees that the discrepancy in an n-node tree will be reduced to at most $\frac{1}{2}n\log n$. Rabani *et al.* [10] analyse a dimension-exchange algorithm in a multiple-token model that achieves optimal discrepancy.

The paper is organised as follows. In Section 2 we formalise the token distribution problem and describe two existing dimension-exchange protocols for this problem. In Section 3 we analyse the performance of these protocols on trees. Our optimal algorithm is presented in Section 4. Due to space limitations many proofs are sketched or omitted.

2 Dimension-Exchange Protocols

The token distribution problem was first posed by Peleg and Upfal [8, 9], and may be stated as follows. Suppose we are given a parallel architecture whose interconnection network is represented by an undirected graph $G = (V,E)$, and a distribution function $\mathsf{load} : V \to \mathbb{N}$, where $\mathsf{load}(v)$ is the number of tokens initially at the node v.

The load of a node v at time t (that is, immediately before step t) is denoted by $\mathsf{load}_t(v)$. We define the (*node*-)*discrepancy* between nodes v and w at time t to be $\Delta_t(v,w) = |\mathsf{load}_t(v) - \mathsf{load}_t(w)|$. The (*edge*-)*discrepancy* of an edge vw at time t is $\Delta_t(vw) = \Delta_t(v,w)$. The (*global*) *maximum* and *minimum load* at time t are $\mathsf{globalMax}_t(G) = \max\{\mathsf{load}_t(v) : v \in V\}$ and $\mathsf{globalMin}_t(G) = \min\{\mathsf{load}_t(v) : v \in V\}$, respectively. The (*global*) *discrepancy* at time t, denoted by $\Delta_t(G)$, is defined to be the maximum node-discrepancy taken over all pairs of nodes; that is, $\Delta_t(G) = \mathsf{globalMax}_t(G) - \mathsf{globalMin}_t(G)$.

The *token distribution problem* is the problem of redistributing the tokens on a given graph so that the global discrepancy of the resulting distribution is minimised. The following lower bound for the time required to solve the token distribution problem on trees is proved in [5] using an elementary bisection-width argument.

Observation 1. *There are instances of the token distribution problem on n-node trees with discrepancy Δ that require $\Omega((\Delta-\delta)\cdot n)$ steps to reduce the discrepancy to δ.*

In this paper we establish upper bounds on the discrepancy of the distribution produced by certain algorithms. With this goal in mind, we now formalise the notion of a distribution which 'cannot be improved' by a particular dimension-exchange algorithm.

Definition 1. For a given dimension-exchange algorithm, we say a distribution of tokens on a graph G is *stable* at some time t, if applying the algorithm leads to a token distribution at some later time $t' > t$ with $t' \equiv t \pmod{\chi}$ such that for every node v, $\mathsf{load}_t(v) = \mathsf{load}_{t'}(v)$. The *maximum stable discrepancy* of a graph G, with respect to a given dimension-exchange algorithm, is the maximum $\delta \in \mathbb{N}$ such that there exists a stable distribution on G with global discrepancy δ.

Our first dimension-exchange protocol, called THRESHOLD-2, always sends a token across an edge with discrepancy at least two, and has appeared in [2].

Protocol THRESHOLD-2 (node v, time t)
  if there exists an edge vw of colour $t \bmod \chi$ incident to v **then**
    send the value $\mathsf{load}_t(v)$ to w and receive the value $\mathsf{load}_t(w)$ from w;
    if $\mathsf{load}_t(v) \geq \mathsf{load}_t(w) + 2$ **then** send one token from v to w; **end-if**
  end-if

Theorem 1. *Let G be a connected graph with diameter $\tau(G)$. Given an arbitrary initial distribution of tokens on G, the* THRESHOLD-2 *protocol will determine a distribution on G with discrepancy at most $\tau(G)$.*

Proof Sketch. By considering the potential function $\sum_v \mathsf{load}_t(v)^2$, it can be proved that under the THRESHOLD-2 protocol, the dimension-exchange algorithm will determine a stable distribution. A distribution is stable under the THRESHOLD-2 protocol if and only if every edge has discrepancy at most one. It follows that $\tau(G)$ is an upper bound on the maximum stable discrepancy of G. To construct a stable distribution on G with discrepancy $\tau(G)$, let v be an end-node of a path in G with $\tau(G)$ edges. Set the load of every node w of G to be the graph-theoretic distance from w to v. This distribution is stable and has discrepancy $\tau(G)$. □

Our second dimension-exchange protocol, called THRESHOLD-1, is stated below. This rule differs from THRESHOLD-2 in that a token is sent across an edge with discrepancy one. THRESHOLD-1 was analysed for meshes and tori in [6], and for complete binary trees in [5]. In Section 3 we analyse THRESHOLD-1 for arbitrary trees.

Protocol THRESHOLD-1 (node v, time t)
  if there exists an edge vw of colour t mod χ incident to v **then**
    send the value $\mathsf{load}_t(v)$ to w and receive the value $\mathsf{load}_t(w)$ from w;
    if $\mathsf{load}_t(v) \geq \mathsf{load}_t(w) + 1$ **then** send one token from v to w; **end-if**
  end-if

Note that this protocol is different to that of Rabani *et al.* [10] in two respects. Firstly THRESHOLD-1 is in the single-token model. Secondly, the protocol in [10] assumes that each edge has a fixed orientation, and in any token exchange an excess token (if it exists) follows the direction of the edge.

3 Analysis of the THRESHOLD-1 Protocol

In this section we provide a number-theoretic method for determining the maximum stable discrepancy of a given tree under the THRESHOLD-1 protocol introduced in Section 2. Let T be a tree whose edges are coloured $0, 1, \ldots, \chi - 1$. (Using depth-first search for example, the edges of a tree with maximum degree D can be coloured with $\chi = D$ colours.) Consider the directed graph T' obtained from T by adding $\chi - \deg(v)$ self-loops to each node v, where $\deg(v)$ is the degree of v in T, and replacing each edge vw of T by two directed arcs $\overrightarrow{vw}$ and $\overrightarrow{wv}$. Every node v of T' has in-degree χ and out-degree χ (where a self-loop counts as both incoming and outgoing). Colour the arcs $\overrightarrow{vw}$ and $\overrightarrow{wv}$ of T' with the same colour as the edge vw in T, and colour the self-loops of T' so that for every colour $c \in \{0, 1, \ldots, \chi - 1\}$, each node has precisely one incoming arc and one outgoing arc coloured with c.

Definition 2. The *observer tour* of T is the cyclic sequence S of the arcs of T' defined by the following rule: if $\overrightarrow{vw}$ is coloured c then the outgoing arc at w coloured $(c+1) \bmod \chi$ is immediately after $\overrightarrow{vw}$ in S. For each edge vw of T, v_w denotes the start-node of the arc $\overrightarrow{vw}$ in the observer tour of T, as shown in Figure 1.

Lemma 1. *For a tree T, the observer tour defines an Eulerian tour of T'.*

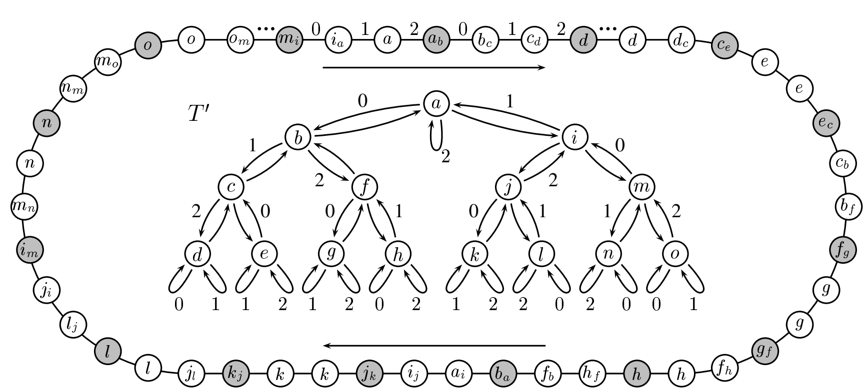

Figure 1: Observer tour of the complete binary tree of height 3.

Proof. Suppose the arc $\overrightarrow{vw}$ is coloured c. The next arc in the observer tour after $\overrightarrow{vw}$ which is also incident with v is $\overrightarrow{wv}$ (otherwise there is a cycle in T). By definition, $\overrightarrow{wv}$ is also coloured c, and thus the arc following $\overrightarrow{wv}$ in the observer tour is the outgoing arc at v coloured $(c+1) \bmod \chi$. Continuing in this manner, if the outgoing arcs at v are ordered $(\overrightarrow{vw_0}, \overrightarrow{vw_1}, \ldots, \overrightarrow{vw_{\chi-1}})$ in the observer tour then $\overrightarrow{vw_c}$, $0 \le c \le \chi - 1$, is coloured c. Therefore, at each node v all arcs incident with v are traversed before the observer tour repeats itself. Hence the observer tour includes all arcs of T' and therefore is an Eulerian tour of T'. □

For each edge vw of T, we denote by $T(v,w)$ the connected subtree of T obtained by removing vw, and containing the node v. Since every node has χ outgoing arcs in T', each node of $T(v,w)$ contributes precisely χ arcs to the directed path on the observer tour from w_v to v_w. Hence we have the following observation, where the number of nodes in a tree T is denoted by $|T|$.

Observation 2. *For each edge vw of a tree T, the number of arcs from w_v to v_w on the observer tour is $\chi \cdot |T(v,w)|$.*

The observer tour defines χ orderings of the nodes of T in the following manner. For each colour $c \in \{0, 1, \ldots, \chi - 1\}$, let $E_c = (\overrightarrow{e_0}, \overrightarrow{e_1}, \ldots, \overrightarrow{e_{n-1}})$ be the cyclic ordering of the arcs in T' coloured with c ordered as they appear in the observer tour. Each node has one outgoing arc in E_c. If $\overrightarrow{e_i}$ and $\overrightarrow{e_j}$ are the outgoing arcs in E_c at distinct nodes v and w, respectively, then we say the *c-gap* from v to w is $\mathsf{gap}_c(v,w) = (j-i) \bmod n$. Since $1 \le \mathsf{gap}_c(v,w) \le n-1$, we call an integer $p \in \{1, 2, \ldots, n-1\}$ a *gap* of T. Clearly $\mathsf{gap}_c(v,w) + \mathsf{gap}_c(w,v) = n$.

The THRESHOLD-1 protocol has the effect of circulating tokens. To see this, consider the action of the THRESHOLD-1 protocol, in the case that a node v of the tree T initially contains one token, and all other nodes contain zero tokens. If xy is an active edge with $\mathsf{load}(x) = 1$ and $\mathsf{load}(y) = 0$ then one token is sent from x

to y. Hence $\mathsf{load}(x)$ becomes zero and $\mathsf{load}(y)$ becomes one. The token will thus follow the sequence of edges starting at v which are coloured $0,1,2,\ldots$; that is, it follows the observer tour. By Lemma 1, the observer tour is an Eulerian tour of T'. Since T' has $\chi \cdot n$ arcs, after $\chi \cdot n$ steps, the token will have traversed the entire tree and will have returned to v.

Definition 3. A *phase* is a sequence of n consecutive rounds; that is, $\chi \cdot n$ consecutive steps. A phase commencing at time $t \equiv c \pmod{\chi}$ is called a *c-phase*.

For our purposes it shall suffice to consider disjoint c-phases for some fixed colour c. Using the potential function $\sum_v \mathsf{load}_t(v)^2$ and the notion of a phase we prove the following result, which implies the maximum stable discrepancy is an upper bound on the final discrepancy of a given distribution under the THRESHOLD-1 protocol.

Lemma 2. *For an arbitrary initial distribution on a tree T, under the* THRESHOLD-1 *protocol, the dimension-exchange algorithm will determine a stable distribution.* □

In order to analyse the effects of the THRESHOLD-1 protocol on the circulation of tokens, it will be convenient to adopt a vantage point which itself circulates through the tree. Associated with each node v of the tree, we consider there to be an 'observer' which at the start of a phase is at v and thereafter follows the observer tour. The sequence of load values encountered by an observer is critical to our analysis. We formalise these notions as follows.

Definition 4. Consider a phase of the dimension-exchange algorithm on a tree T starting at time $t_0 \equiv c \pmod{\chi}$. For each node v of T the *observer* of v at time t, $t_0 \le t < t_0 + \chi \cdot n$, denoted by $\mathsf{obs}_t(v)$, is the node w with $\mathsf{gap}_c(v,w) = t$. We say $\mathsf{obs}(v)$ *is at* w at time t if $\mathsf{obs}_t(v) = w$. (The load of an observer is thus the load of the node where the observer is currently situated.) At a particular time point during the phase, we say an observer is *maximum* (respectively, *minimum*) if the current load of the observer equals $\mathsf{globalMax}_{t_0}(T)$ ($\mathsf{globalMin}_{t_0}(T)$); that is, the maximum (minimum) load at the *start* of the phase.

Figure 2 provides an example of a stable token distribution. The colour of each edge and the load of each node after each step is indicated. There are two maximum observers and one minimum observer, each of which remain maximum or minimum observers throughout the phase.

We now describe how to determine the maximum stable discrepancy of a tree $T = (V,E)$ under the THRESHOLD-1 protocol. Suppose there is a stable distribution on T at time $t \equiv c \pmod{\chi}$. In Lemma 3 below we show that if $\mathsf{gap}_c(v,w) = |T(x,y)|$ for some pair of nodes v, w and some edge xy, then the node-discrepancy $\Delta_t(v,w) \le 1$. We therefore define the *stable gaps for discrepancy 1* as follows:

$$\mathsf{SG}_1(T) = \{|T(x,y)|, |T(y,x)| : xy \in E\} \ .$$

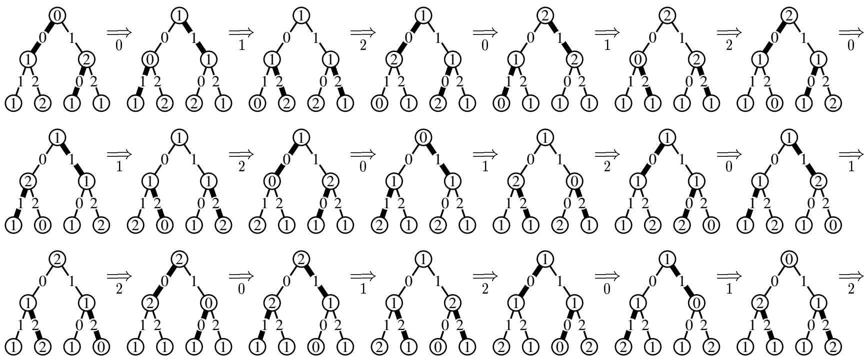

Figure 2: One phase of THRESHOLD-1 on a stable distribution.

Since $|T(x,y)| + |T(y,x)| = n$, if $p \in \mathsf{SG}_1(T)$ then $n - p \in \mathsf{SG}_1(T)$. It follows from Observation 2 that the stable gaps for discrepancy 1 determine which observers meet at an active edge during a phase; see Figure 3.

Lemma 3. *Under the action of the* THRESHOLD-1 *protocol on a tree T, two observers $\mathsf{obs}(v)$ and $\mathsf{obs}(w)$ in a particular c-phase are at end-nodes of a common active edge during this phase if and only if $\mathsf{gap}_c(v,w) \in \mathsf{SG}_1(T)$.* □

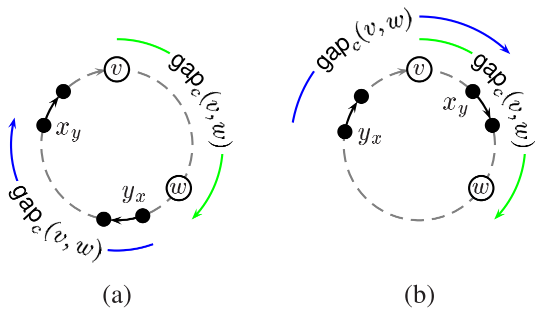

Figure 3: Relative positions of v, w, and the edge xy in the c-ordering starting at v.

In a stable distribution, whenever two observers meet at an active edge, their discrepancy must be at most one. Since Lemma 3 characterises when two observers will meet at an active edge, we have a necessary condition for a distribution to be stable. The next result asserts that this condition is sufficient.

Lemma 4. *A distribution on a tree T is stable under the* THRESHOLD-1 *protocol at some time $t \equiv c \pmod{\chi}$ if and only if every pair v, w of nodes of T with $\mathsf{gap}_c(v,w) \in \mathsf{SG}_1(T)$ has node-discrepancy $\Delta_t(v,w) \leq 1$.* □

For any stable distribution at time $t \equiv c \pmod{\chi}$, we have shown that if two nodes have c-gap in $\mathsf{SG}_1(T)$, then their node-discrepancy must be at most one. If two nodes have a c-gap of $(p+q) \bmod n$, for some $p,q \in \mathsf{SG}_1(T)$, then in a stable distribution their node-discrepancy must be at most two. We therefore define the *stable gaps for discrepancy i* ($i \geq 2$) as follows:

$$\mathsf{SG}_i(T) = \left\{ \Big(\sum_{j=1}^{k} p_j \Big) \bmod n : p_j \in \mathsf{SG}_1(T), 1 \leq k \leq i \right\} .$$

We define the *stability* of a gap p of T to be

$$\mathsf{stability}(p) = \min\{i \geq 1 : p \in \mathsf{SG}_i(T)\}$$

For each gap p, if $p \in \mathsf{SG}_i(T)$ then $n-p \in \mathsf{SG}_i(T)$, and hence $\mathsf{stability}(p) = \mathsf{stability}(n-p)$. We now provide a second characterisation of stable distributions under the THRESHOLD-1 protocol in terms of the stability of gaps.

Theorem 2. *Let T be a tree whose edges are coloured $0,1,\dots,\chi-1$. Under the* THRESHOLD-1 *protocol, a distribution on T is stable at time $t \equiv c \pmod{\chi}$ if and only if for all pairs of nodes v, w of T, the node-discrepancy $\Delta_t(v,w) \leq \mathsf{stability}(\mathsf{gap}_c(v,w))$.*

Proof. ($\Longleftarrow$) Suppose that for all pairs of nodes v, w of T the node-discrepancy $\Delta_t(v,w) \leq \mathsf{stability}(\mathsf{gap}_c(v,w))$. Then for all pairs of nodes v, w of T with $\mathsf{gap}_c(v,w) \in \mathsf{SG}_1(T)$, the node-discrepancy $\Delta_t(v,w) \leq 1$. By Lemma 4, the distribution is stable.

($\Longrightarrow$) We prove the 'only if' part of this result by induction on i with the following induction hypothesis: *If a distribution on T is stable at time $t \equiv c \pmod{\chi}$ under the* THRESHOLD-1 *protocol, then for all pairs of nodes v,w of T with $\mathsf{stability}(\mathsf{gap}_c(v,w)) = i$, the node-discrepancy $\Delta_t(v,w) \leq i$.*

The basis of the induction with $i = 1$ is the 'only if' assertion in Lemma 4. Let $i \geq 2$, and assume that the induction hypothesis is true for values less than i. Assume, to the contrary, that there is a stable distribution on T at time $t \equiv c \pmod{\chi}$ such that for some nodes v and w with $\mathsf{stability}(\mathsf{gap}_c(v,w)) = i$, the node-discrepancy $\Delta_t(v,w) \geq i+1$. Thus $\mathsf{gap}_c(v,w) \in \mathsf{SG}_i(T) \setminus \mathsf{SG}_{i-1}(T)$, and hence

$$\mathsf{gap}_c(v,w) = \left(\sum_{j=1}^{i} p_j \right) \bmod n \quad , \tag{1}$$

with $p_j \in \mathsf{SG}_1(T)$. Let x be the node with $\mathsf{gap}_c(v,x) = p_i$. Observe that $\mathsf{gap}_c(v,x) + \mathsf{gap}_c(x,w) \equiv \mathsf{gap}_c(v,w) \pmod{n}$. Hence $\mathsf{gap}_c(x,w) \equiv \mathsf{gap}_c(v,w) - p_i \pmod{n}$, and by (1),

$$\mathsf{gap}_c(x,w) = \left(\sum_{j=1}^{i-1} p_j \right) \bmod n \quad .$$

Thus $\mathsf{gap}_c(x,w) \in \mathsf{SG}_{i-1}(T)$, and by the induction hypothesis, $\Delta_t(x,w) \leq i-1$. Since $p_i \in \mathsf{SG}_1(T)$, by the basis of the induction, $\Delta_t(v,x) \leq 1$. By the triangle inequality, $\Delta_t(v,w) \leq \Delta_t(v,x) + \Delta_t(x,w) \leq 1 + (i-1) = i$, which contradicts our initial assumption. □

For an n-node tree T, we define

$$\mathsf{MSD}(T) = \min\{i \geq 1 : \mathsf{SG}_i(T) = \{1,2,\ldots,n-1\}\} \ .$$

Equivalently, $\mathsf{MSD}(T)$ is the maximum stability taken over all gaps of T. Note that $\mathsf{MSD}(T)$ is not defined with respect to a particular edge-colouring.

Theorem 3. *The maximum stable discrepancy of a tree T under the* THRESHOLD-1 *protocol is* $\mathsf{MSD}(T)$.

Proof Sketch. For every gap p of T there exist pairs of nodes v, w with $\mathsf{gap}_c(v,w) = p$. It follows from Theorem 2 that there is no stable distribution on T with greater global discrepancy than $\mathsf{MSD}(T)$. We now construct, for an arbitrary time t, a distribution on T with discrepancy $\mathsf{MSD}(T)$ which is stable at time t. Let s be an arbitrary node of T. Set $\mathsf{load}(s) \leftarrow 0$, and for every other node v, set $\mathsf{load}(v) \leftarrow \mathsf{stability}(\mathsf{gap}_c(s,v))$ where $t \equiv c \pmod{\chi}$. Since $\mathsf{SG}_{i-1}(T) \subseteq \mathsf{SG}_i(T)$ for every $i \geq 2$, the discrepancy of this distribution is $\mathsf{MSD}(T)$. It can be shown that this distribution is stable at time t, and therefore the maximum stable discrepancy is $\mathsf{MSD}(T)$. □

For every n-node tree T, $1 \in \mathsf{SG}_1(T)$. It follows that the maximum stable discrepancy of T under the THRESHOLD-1 protocol is at most $n/2$. Let S_k be the k-star; that is, the tree with k edges all incident to a single node. Then $\mathsf{SG}_1(S_k) = \{1\}$. It follows that $\mathsf{MSD}(S_k) = \lfloor (k+1)/2 \rfloor = \lfloor n/2 \rfloor$, and the above observation is tight for S_k. By Theorem 1, the maximum stable discrepancy of S_k under the THRESHOLD-2 protocol is 2. Therefore the THRESHOLD-2 protocol is superior to the THRESHOLD-1 protocol for star architectures. Conversely, by Theorem 1, the n-node path P_n has maximum stable discrepancy of $n-1$ under the THRESHOLD-2 protocol, and since $\mathsf{SG}_1(P_n) = \{1,2,\ldots,n-1\}$, Theorem 3 implies that P_n has maximum stable discrepancy of 1 under the THRESHOLD-1 protocol. Therefore for paths, THRESHOLD-1 is superior to THRESHOLD-2.

We now establish a logarithmic upper bound on the maximum stable discrepancy of an arbitrary tree under the THRESHOLD-1 protocol. The CONSTRUCT SUBGRAPH algorithm to follow, given a tree T and gap p of T, determines a connected subtree α with p nodes. It does so by building up the subgraph α from a single node, and maintaining a connected subgraph β, disjoint from α, of candidate nodes for inclusion into α such that β has one node adjacent to a node in α. We implicitly associate the subgraphs α and β with the sets of nodes which respectively induce them. The algorithm makes use of the *centroid* of a tree, defined as follows. For each node v of a tree T, let $C(v) = \max\{|S| :$

S is a connected subgraph of $T \setminus \{v\}\}$. A centroid is any node v of T for which $C(v)$ is minimum. Kang and Ault [7] show that if v is a centroid of T, then $C(v) \leq |T|/2$.

Algorithm CONSTRUCT SUBGRAPH (gap p of a tree $T = (V,E)$)
choose a leaf-node l of T;
set $\alpha \leftarrow \{l\}$, $\beta \leftarrow V \setminus \{l\}$;
while $|\alpha| < p$ **do**
let v be a centroid node of β with $d = \deg(v)$;
let $\beta_1, \beta_2, \ldots, \beta_{\deg(v)}$ be the connected subgraphs of $\beta \setminus \{v\}$ such that
β_1 contains a node adjacent to a node in α, or
if v is adjacent to a node in α then $\beta_1 = \emptyset$;
(refer to Figure 4)
Case 1: if $|\alpha| + |\beta_1| > p$ **then** set $\beta \leftarrow \beta_1$; **else**
Case 2: if $|\alpha| + |\beta_1| = p$ **then** set $\alpha \leftarrow \alpha \cup \beta_1$; **else**
Case 3: if $|\alpha| + |\beta_1| < p$ **then**

$$\text{set } i \leftarrow \min\left\{ j \geq 1 : |\alpha| + 1 + \sum_{k=1}^{j} |\beta_k| > p \right\};$$

$$\text{set } \alpha \leftarrow \alpha \cup \{v\} \cup \bigcup_{j=1}^{i-1} \beta_j \text{ and } \beta \leftarrow \beta_i;$$

end-if
end-while

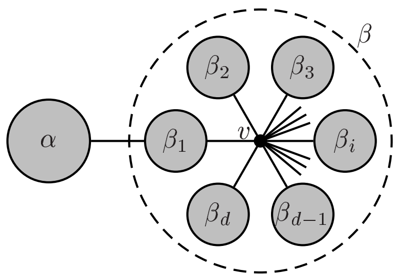

Figure 4: The subgraphs $\beta_1, \beta_2, \ldots, \beta_d$ associated with a centroid node v of β.

Lemma 5. *Let p be a gap of an n-node tree T with maximum degree D, and let $M = \min\{1 + (D-2)\lceil \log_2 n \rceil, \lfloor \frac{D+1}{2} \lceil \log_2 n \rceil \rfloor\}$. Then $p \in \mathsf{SG}_M(T)$.*

Proof Sketch. Given a gap p of T, the algorithm CONSTRUCT SUBGRAPH(p) determines a connected subtree α with p nodes, such that if $v_1, v_2, \ldots, v_r$ are the vertices in α incident to an edge whose other endpoint is not in α then

$r \le \lceil \log_2 n \rceil$. For each vertex v_i, $1 \le i \le r$, let $w_{i,1}, w_{i,2}, \ldots, w_{i,c_i}$ be the nodes adjacent to v_i which are not in α, and let $w_{i,c_i+1}, w_{i,c_i+2}, \ldots, w_{i,d_i}$ be the nodes adjacent to v_i which are in α, where $d_i = \deg(v_i)$. For $1 \le j \le d_i$, let $\gamma_{i,j} = |T(w_{i,j}, v_i)|$. Each $\gamma_{i,j} \in \mathsf{SG}_1(T)$ and thus $n - \gamma_{i,j} \in \mathsf{SG}_1(T)$. It can be shown that $p = \sum_{i,j}(n - \gamma_{i,j}) \bmod n$. Let $M_1 = \sum_{i=1}^{r} c_i$. Then p is the sum modulo n of M_1 terms of $\mathsf{SG}_1(T)$, and hence $p \in \mathsf{SG}_{M_1}(T)$. It can be shown that $M_1 \le 1 + \sum_{i=1}^{r}(d_i - 2) \le 1 + (D-2)r$. Hence $p \in \mathsf{SG}_{1+(D-2)\lceil \log_2 n \rceil}(T)$. This establishes the first part of the result. We can also express $n - p$ as the sum of terms in $\mathsf{SG}_1(T)$ as follows.

$$n - p \;=\; (n - r) + \sum_{i=1}^{r} \sum_{j=c_i+1}^{d_i} (n - \gamma_{i,j}) \bmod n \;\; .$$

$r = r \cdot 1$ is the sum of r terms of $\mathsf{SG}_1(T)$. Hence $n - r$ is the sum modulo n of at most r terms of $\mathsf{SG}_1(T)$. Since each $n - \gamma_{i,j} \in \mathsf{SG}_1(T)$, it follows that $n - p$ is the sum modulo n of at most

$$r + \sum_{i=1}^{r}(d_i - c_i) \;=\; \sum_{i=1}^{r}(1 + d_i - c_i) \;\le\; \sum_{i=1}^{r}(D + 1 - c_i) \;=\; r(D+1) - M_1$$

terms of $\mathsf{SG}_1(T)$. Hence $n - p \in \mathsf{SG}_{r(D+1)-M_1}(T)$ and thus $p \in \mathsf{SG}_{r(D+1)-M_1}(T)$. Clearly $\min\{M_1, r(D+1) - M_1\} \le \lfloor r(D+1)/2 \rfloor$. Hence $p \in \mathsf{SG}_{\lfloor r(D+1)/2 \rfloor}(T)$. Since $r \le \lceil \log_2 n \rceil$, $p \in \mathsf{SG}_{\lfloor (D+1)\lceil \log_2 n \rceil /2 \rfloor}(T)$. This establishes the second part of the result. □

By Theorem 3, and Lemmata 2 and 5 we obtain our main result of this section.

Theorem 4. *Given an arbitrary initial distribution of tokens on an n-node tree with maximum degree D, under the* THRESHOLD-1 *protocol, the final distribution has discrepancy at most* $\min\{\lfloor \frac{n}{2} \rfloor, 1 + (D-2)\lceil \log_2 n \rceil, \lfloor \frac{D+1}{2} \lceil \log_2 n \rceil \rfloor\}$. □

Houle, Tempero, and Turner [5], who introduced the THRESHOLD-1 protocol for trees, provided analysis only in the case of the complete binary tree. The following upper bound on the maximum stable discrepancy of the complete k-ary tree of height h ($k \ge 1, h \ge 1$) matches the bound in [5] for $k = 2$ (up to an additive constant of 1).

Lemma 6. *Under the* THRESHOLD-1 *protocol, the maximum stable discrepancy of the complete k-ary tree of height h,* $T_{h,k}$ $(k \ge 1, h \ge 1)$ *is at most* $\min\{(k-1)h + 1, (k+2)(h+1)/2\}$.

Proof Sketch. The number of nodes in $T_{h,k}$ is $|T_{h,k}| = \frac{k^{h+1}-1}{k-1}$. The upper bound of $(k-1)h+1$ is proved by induction on the height h of $T_{h,k}$ (for fixed $k \ge 1$) with the inductive hypothesis: *Every gap p of* $T_{h,k}$ *is the sum of at most* $(k-1)h+1$ *terms in* $\{|T_{j,k}| : 0 \le j \le h-1\}$. Since each $|T_{j,k}| \in \mathsf{SG}_1(T_{h,k})$, the result follows.

The upper bound of $(k+2)(h+1)/2$ is proved as follows. Consider the CONSTRUCT SUBGRAPH algorithm applied to a complete k-ary tree of height h. In the first iteration, the subtree β is a complete k-ary tree with a leaf-node removed. Thereafter β is a complete k-ary tree with progressively smaller height. Hence $\beta_i = (|\beta|-1)/k$, and therefore the algorithm terminates in $\lceil \log_k n \rceil = h+1$ iterations. Since the maximum degree is $k+1$, it follows using the analysis of Lemma 5 that the maximum stable discrepancy is at most $(k+2)(h+1)/2$. □

We conjecture that for all $h \geq 1$ and $k \geq 1$, the maximum stable discrepancy of $T_{h,k}$ under the THRESHOLD-1 protocol is $\lfloor (k-1)h/2 \rfloor$ or $\lfloor (k-1)h/2 \rfloor + 1$, which has been confirmed by directly calculating $\mathsf{MSD}(T_{h,k})$ for the complete binary tree $T_{h,2}$ with $h \leq 18$, and for $T_{h,k}$ with $1 \leq h \leq 6$ and $1 \leq k \leq 6$.

4 An Optimal Algorithm

In this section we present the algorithm DISCREPANCY-1, which reduces the discrepancy of an arbitrary distribution to at most one. This algorithm depends on additional information being stored at each node. In particular, each node stores the maximum number of tokens at that node during certain time periods, and we assume that each node has knowledge of the total number of nodes in the tree.

A distribution with relatively large discrepancy can be stable under the THRESHOLD-1 protocol, since maximum and minimum observers may never meet at an active edge during a phase. In the USELOCALMAX protocol below, if a node with maximum load is incident to an active edge with discrepancy one, then no token is sent across the edge — in effect, the THRESHOLD-2 protocol is running at nodes with maximum load. Note that the choice of 'freezing' nodes with maximum loads is arbitrary; we could instead 'freeze' nodes with minimum load. Based purely on local information, however, each node has no way of knowing if its current load is a global maximum. For each node v, the algorithm stores $\mathsf{localMax}(v)$, which can be considered a 'local approximation' to the current global maximum. We shall describe later how $\mathsf{localMax}(v)$ is determined.

Protocol USELOCALMAX (node v, time t)
    if there exists an edge vw of colour t mod χ incident to v **then**
        send the value $\mathsf{load}_t(v)$ to w and receive the value $\mathsf{load}_t(w)$ from w;
        if $\mathsf{load}_t(v) \geq \mathsf{load}_t(w) + 2$ or
                $(\mathsf{load}_t(v) = \mathsf{load}_t(w) + 1$ and $\mathsf{load}_t(v) \neq \mathsf{localMax}(v))$ **then**
            send one token from v to w;
        end-if
    end-if

Algorithm DISCREPANCY-1 below runs in *cycles*, each composed of an *A-phase* followed by a *B-phase*. During the A-phase, we use the THRESHOLD-1 protocol, and the local information localMax(v) is updated so that at the end of the A-phase, for each node v, localMax(v) is the maximum number of tokens stored at v at any one time during the A-phase. In the B-phase we apply the USELOCALMAX protocol, using the local information gained during the previous A-phase.

Algorithm DISCREPANCY-1 (node v, time t)
  localMax(v)←load(v);
A: **for** $j = 1$ **to** $\chi \cdot n$ **do**
    apply THRESHOLD-1(v,t);
    if load$_t(v)$ $>$ localMax(v) **then** localMax(v) $\leftarrow$ load$_t(v)$;
    $t \leftarrow t+1$;
  end-for
B: **for** $j = 1$ **to** $\chi \cdot n$ **do**
    apply USELOCALMAX(v,t);
    $t \leftarrow t+1$;
  end-for

Theorem 5. *For an arbitrary initial distribution on a tree T, repeated application of the algorithm* DISCREPANCY-1 *will decrease the discrepancy to at most one. Furthermore, for bounded degree trees, the rate of convergence is asymptotically optimal in the worst case.*

Proof Sketch. We show that if the discrepancy has not decreased by the end of the A-phase, then during the B-phase either the global maximum decreases or the global minimum increases. If during the A-phase the discrepancy decreases, then we are done. We now assume that during the A-phase the discrepancy has not decreased. Clearly there is at least one maximum observer which has survived the A-phase. Since such an observer traverses every node of the tree, at the end of the A-phase, for every node v of T, localMax(v) is the global maximum at the start of the A-phase.

If at the end of the B-phase the global maximum has decreased, then the global discrepancy has decreased, and we are done. Otherwise, we show that since a maximum 'does not move' during the B-phase, the global minimum must increase during the B-phase. Hence the global discrepancy has decreased by at least one.

The number of steps to reduce the discrepancy of a given distribution to at most one is no more than $2(\Delta_0(T)-1)\cdot\chi n$, which by the lower bound of Observation 1 is asymptotically optimal for trees with bounded degree. $\square$

References

[1] G. CYBENKO, Dynamic load balancing for distributed memory multiprocessors. *J. Parallel Distrib. Comput.*, **7**:279–301, 1989.

[2] B. GHOSH, F. T. LEIGHTON, B. M. MAGGS, S. MUTHUKRISHNAN, C. G. PLAXTON, R. RAJARAMAN, A. W. RICHA, R. E. TARJAN, AND D. ZUCKERMAN, Tight analyses of two local load balancing algorithms. *SIAM J. Comput.*, **29(1)**:29–64, 1999.

[3] B. GHOSH AND S. MUTHUKRISHNAN, Dynamic load balancing by random matchings. *J. Comput. System Sci.*, **53(3)**:357–370, 1996.

[4] S. H. HOSSEINI, B. LITOW, M. MALKAWI, J. MCPHERSON, AND K. VAIRAVAN, Analysis of graph coloring based distributed load balancing algorithm. *J. Parallel Distrib. Comput.*, **10**:160–166, 1990.

[5] M. E. HOULE, E. TEMPERO, AND G. TURNER, Optimal dimension-exchange token distribution on complete binary trees. *Theoret. Comput. Sci.*, **220(2)**:363–376, 1999.

[6] M. E. HOULE AND G. TURNER, Dimension-exchange token distribution on the mesh and the torus. *Parallel Comput.*, **24(2)**:247–265, 1998.

[7] A. N. C. KANG AND D. A. AULT, Some properties of a centroid of a free tree. *Inform. Process. Lett.*, **4(1)**:18–20, 1975.

[8] D. PELEG AND E. UPFAL, The generalized packet routing problem. *Theoret. Comput. Sci.*, **53**:218–293, 1987.

[9] D. PELEG AND E. UPFAL, The token distribution problem. *SIAM J. Comput.*, **18(2)**:229–243, 1989.

[10] Y. RABANI, A. SINCLAIR, AND R. WANKA, Local divergence of Markov chains and the analysis of iterative load-balancing schemes. In *Proc. 39th Annual Symposium on Foundations of Computer Science (FOCS '98)*, pp. 694–703, IEEE, 1998.

[11] V. G. VIZING, On an estimate of the chromatic class of a p-graph. *Diskret. Analiz No.*, **3**:25–30, 1964.

Michael E. Houle is with IBM Research, Tokyo Research Laboratory, Japan, and the School of Information Technologies, The University of Sydney, Australia (on leave). E-mail: `meh@trl.ibm.co.jp`

Antonios Symvonis is with the Department of Mathematics, University of Ioannina, Ioannina, Greece. E-mail: `symvonis@cc.uoi.gr`

David R. Wood is with the School of Computer Science, Carleton University, Ottawa, Canada. E-mail: `davidw@scs.carleton.ca`