Available online at www.sciencedirect.com
SCIENCE DIRECT®
J. Parallel Distrib. Comput. 64 (2004) 591–605
Journal of Parallel and Distributed Computing
http://www.elsevier.com/locate/jpdc

Dimension-exchange algorithms for token distribution on tree-connected architectures ☆

Michael E. Houle,[a] Antonios Symvonis,[b] and David R. Wood[c,*,1]

[a] *National Institute of Informatics, Tokyo, Japan*
[b] *Department of Mathematics, National Technical University of Athens, Athens, Greece*
[c] *School of Computer Science, Carleton University, 1125 Colonel By Drive, Ottawa, Canada*

Received 14 March 2001; revised 30 September 2003

Abstract

Load balancing on a multi-processor system involves redistributing tasks among processors so that each processor has roughly the same amount of work to perform. The *token-distribution* problem is a static variant of the load balancing problem for the case in which the workloads in the system cannot be divided arbitrarily; that is, where each token represents an atomic element of work. A scalable method for distributing tokens over a parallel architecture is the so-called *dimension-exchange* approach. Our results include improved analysis of two existing dimension-exchange algorithms for token distribution on arbitrary graphs and on arbitrary trees, respectively. In particular, we establish a logarithmic upper bound on the discrepancy of the resulting distribution when the second algorithm is applied to an arbitrary initial distribution on a tree. We then present a new dimension-exchange algorithm for token distribution on trees, which assuming each node knows the number of nodes in the tree, determines a 'perfectly balanced' distribution. Furthermore, the rate of convergence is worst-case optimal for trees of bounded degree. Note that an algorithm for token-distribution on trees is applicable to arbitrary architectures, since the algorithm can be applied on a spanning tree of any given connected graph.

Keywords: Load balancing; Token distribution; Dimension-exchange; Tree

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 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.

In this paper it is assumed that each token requires only a constant amount of time to be sent from one PE to an adjacent PE, and that no tokens are created or destroyed before the redistribution is complete. We assume that each PE has facilities for synchronous single-port communication. Under this assumption, the PEs are connected to their neighbours by bi-directional communication links, and may send and receive at most one message at any one time. This model is considerably weaker than the 'multiport' model, where concurrent communication to all the neighbours is allowed. The multiport model has been employed for token distribution in [9,13,17,22,30]. Note that bi-directional communication links can be simulated by a constant number of

☆ An extended abstract of this paper was presented at the 9th International Colloquium on Structural Information and Communication Complexity (SIROCCO '02).

*Corresponding author. Fax: +1-613-520-4334.
E-mail addresses: meh@nii.ac.jp (M.E. Houle), symvonis@math.ntua.gr (A. Symvonis), davidw@scs.carleton.ca (D.R. Wood).

[1] Research supported by the Natural Sciences and Engineering Research Council of Canada.

0743-7315/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.jpdc.2004.03.011

communication steps over uni-directional links. Our results therefore apply in this weaker model with constant factor slow-down. We choose the bi-directional model for ease of explanation. We model the interconnection network of the parallel architecture by a connected simple undirected graph, whose nodes correspond to PEs and edges correspond to communication links.

1.1. Dimension-exchange algorithms

There are many data distribution methods that achieve a balanced token distribution by gathering and making use of a certain amount of global information [4,5,7,8,20,28]. 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 for token distribution in the single-port model that requires no such global information is the so-called *dimension-exchange* method. To implement a dimension-exchange algorithm on a particular parallel architecture, the edges of the corresponding graph are coloured such that no two edges incident to a common node receive the same colour. (The classical result of Vizing [29] states that a simple graph with maximum degree D has such an edge-colouring with $D+1$ 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 of colour $t \pmod{\chi}$ incident to v **then**
      let vw be this (unique) edge;
      exchange information on loads between v and w;
      compare the loads of v and w according to some protocol;
      if required, send a token(s) from v to w or receive a 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.

Definition 1. In each parallel step, those edges of the colour under consideration are said to be *active*. A sequence of χ consecutive parallel steps is called a *round*. (During a round every edge is active exactly once.)

Due to their simplicity and scalability, many researchers have studied the applicability of dimension-exchange techniques to load balancing problems. Cybenko [6] proposed a dimension-exchange algorithm for the d-dimensional hypercube under the assumption that the load in each node is *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 between the two nodes involved, then after d iterations the difference between the maximum and minimum load over all nodes of the network (called the *discrepancy*) would be the minimum possible.

This original work prompted a steady stream of research into the analysis of dimension-exchange algorithms. Hosseini et al. [14] demonstrated that, for infinitely-divisible loads, Cybenko's analysis could be generalised to arbitrary χ-colourable networks. Xu and Lau [31,32] and Litow [19] extended the work in [14] by showing that for the chain, ring, mesh and toroidal mesh topologies, the rate at which the discrepancy converged to zero could be optimised by altering the ratio with which infinitely-divisible loads were locally balanced.

To date, a large body of results exist detailing the performance of the dimension-exchange approach over infinitely-divisible loads. On the other hand, less is known concerning dimension-exchange under the more realistic assumption of *finitely-divisible* loads; that is, loads representable as a set of tokens.

Ranka et al. [27] studied the operation of Cybenko's algorithm empirically for the d-dimensional hypercube assuming finitely-divisible loads. They observed that the discrepancy would eventually fall to at most d. Hosseini et al. [14] and Plaxton [25] independently confirmed this observation by providing algorithms which, after d steps, reduced the discrepancy to at most d.

Ghosh and Muthukrishnan [11] and Ghosh et al. [10] studied the performance of a randomised dimension-exchange algorithm for token distribution on arbitrary graphs (as well as a deterministic algorithm which transfers tokens across all edges simultaneously). Their algorithm determines a random matching at each parallel step, as opposed to cycling through the edges with respect to a fixed edge-colouring.

Houle and Turner [16] proposed and analysed a dimension-exchange algorithm for the two-dimensional mesh and torus. The algorithm was shown to reduce 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 et al. [15] for token distribution on the complete binary tree. They showed that the discrepancy converges to at most the height of the tree, and the rate of this convergence is optimal in the worst case.

Load balancing from a more applied viewpoint has also been widely studied; see [1–3,12,21,26] for example. Note that this list is far from exhaustive. The interested reader should refer to [21,26] and the references therein.

1.2. Our results

We now describe the contributions of this paper, and how they improve upon existing results in the literature.

The contributions of this paper are three-fold. Firstly, we analyse a well-known dimension-exchange protocol, and show that for an arbitrary initial distribution of tokens on a graph, the algorithm reduces the discrepancy of loads to at most the diameter of the graph. Secondly, we provide a new analysis of the dimension-exchange protocol introduced in [15,16] 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 ($k\geqslant 1$, $h\geqslant 1$) to at most

$$\min\{(k-1)h+1, (k+2)(h+1)/2\}.$$

This result generalises the above-mentioned result of Houle et al. [15] for the complete binary tree to the case of any complete k-ary tree.

Thirdly, we present a new dimension-exchange algorithm for trees, which assuming that each node has knowledge of the number of nodes in the tree, reduces the discrepancy of an arbitrary token distribution to at most one. For trees of bounded degree, the rate of convergence is shown to be optimal in the worst-case. This is the first local dimension-exchange algorithm for the token distribution problem on tree-connected architectures that achieves optimal discrepancy. Note that an algorithm for token-distribution on trees is applicable to arbitrary architectures, since the algorithm can be applied on a spanning tree of a given connected graph.

The paper is organised as follows. In Section 2, we formalise the token distribution problem and describe the two existing dimension-exchange protocols for this problem. In Sections 3 and 4 we analyse the performance of these protocols on graphs and trees, respectively. Our algorithm for reducing the discrepancy of an arbitrary initial distribution on a tree to at most one is presented in Section 5. Conclusions and open problems are presented in Section 6.

2. Protocols for dimension-exchange algorithms

The token distribution problem was first posed by Peleg and Upfal [23,24], 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 load: $V\longrightarrow\mathbb{N}$ where $\mathrm{load}(v)$ is the number of tokens initially at the node v.

The load of a node v at time t (that is, immediately before time step t) is denoted by $\mathrm{load}_t(v)$. We define the (*node-*)*discrepancy* between nodes v and w at time t, denoted by $\Delta_t(v,w)$, to be

$$\Delta_t(v,w)=|\mathrm{load}_t(v)-\mathrm{load}_t(w)|.$$

The (*edge-*)*discrepancy* of an edge vw at time t, denoted by $\Delta_t(v,w)$, is the node-discrepancy between v and w at time t. The (*global*) *maximum* and *minimum load* at time t are $\mathrm{globalMax}_t(G)=\max\{\mathrm{load}_t(v): v\in V\}$ and $\mathrm{globalMin}_t(G)=\min\{\mathrm{load}_t(v): v\in V\}$, 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,

$$\Delta_t(G)=\mathrm{globalMax}_t(G)-\mathrm{globalMin}_t(G).$$

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 [15] 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 $\Omega((\Delta-\delta)\cdot n)$ parallel 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, $\text{load}_t(v) = \text{load}_{t'}(v)$. 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.

For each of the dimension-exchange protocols introduced in this paper, an arbitrary initial distribution always converges to a stable distribution. Hence the maximum stable discrepancy is an upper bound on the final discrepancy of the distribution produced by a particular protocol.

Our first dimension-exchange protocol, called THRESHOLD-2, always sends a token across an edge with discrepancy at least two, and has appeared in [10].

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

Note that with the THRESHOLD-2 protocol running synchronously at both v and w, a token sent from v will be received at w and vice versa. In Section 3, we prove that for a graph G with diameter d, the maximum stable discrepancy of G under the THRESHOLD-2 protocol is at most d, and hence, given an arbitrary initial distribution of tokens on G, the THRESHOLD-2 protocol will determine a distribution on G with discrepancy at most d.

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 [16], and for complete binary trees in [15]. In Section 4, we analyse THRESHOLD-1 for arbitrary trees.

Protocol THRESHOLD-1 (node v, time t)
if there exists an edge of colour $t \pmod{\chi}$ incident to v
then
let vw be this (unique) edge;
send the value $\text{load}_t(v)$ to w and receive the value $\text{load}_t(w)$ from w;
if $\text{load}_t(v) \geqslant \text{load}_t(w) + 1$ **then** send one token from v to w;
end-if

We now make some elementary observations common to THRESHOLD-2 and THRESHOLD-1. Consider the potential function $\sum_v \text{load}(v)^2$ under the action of the THRESHOLD-2 or the THRESHOLD-1 protocol. If the discrepancy of an active edge xy is at most one, then the discrepancy of xy is unchanged by the application of either protocol, and hence $\sum_v \text{load}(v)^2$ is unchanged. If the discrepancy of an active edge xy is at least two, then under either THRESHOLD-2 or THRESHOLD-1, one token is moved from the node with greater load to the node with lesser load. It is easily seen that in this case, $\text{load}(x)^2 + \text{load}(y)^2$ decreases by at least two. The following observation immediately follows.

Observation 2. *For an arbitrary distribution on a graph, under the* THRESHOLD-2 *or the* THRESHOLD-1 *protocol, the function* $\sum_v \text{load}_t(v)^2$ *is non-increasing with* t.

This enables us to prove the following assertion concerning stable distributions.

Lemma 2. *Suppose a distribution on a graph* G *is stable at time* t *under the* THRESHOLD-2 *or the* THRESHOLD-1 *protocol. Then at every time step after* t, *the discrepancy of every active edge is at most one.*

Proof. Suppose to the contrary that a distribution on G is stable at time t_0, and at some time $t_1 > t_0$ there is an active edge xy with discrepancy at least two. We can assume without loss of generality that t_1 is the first time after t_0 that xy is active with discrepancy at least two. As stated earlier, $\text{load}(x)^2 + \text{load}(y)^2$ will decrease by at least two at time t_1. By Observation 2, $\sum_v \text{load}(v)^2$ is non-increasing with time. Thus

$$\sum_v \text{load}_{t_1+1}(v)^2 < \sum_v \text{load}_{t_0}(v)^2 \ . \tag{1}$$

Now applying the definition of a stable distribution, there is some time $t_2 > t_0$ with $t_2 \equiv t \pmod{\chi}$ such that for every node v, $\text{load}_{t_0}(v) = \text{load}_{t_2}(v)$. Therefore, by (1) we have

$$\sum_v \text{load}_{t_1+1}(v)^2 < \sum_v \text{load}_{t_0}(v)^2 = \sum_v \text{load}_{t_2}(v)^2. \tag{2}$$

After t_2 the algorithm will repeat the same movement of tokens as carried out between t_0 and t_2. Hence, the first time xy is active with discrepancy at least two is before t_2; that is, $t_1 < t_2$. However, Observation 2 asserts that $\sum_v \text{load}(v)^2$ is non-increasing with time, which contradicts (2), as required. □

The next observation affirms that the global discrepancy is also non-increasing with time. An elementary proof is given in [15].

Observation 3. *For an arbitrary distribution on a graph, the maximum load is non-increasing and the minimum load is non-decreasing with time, under the* THRESHOLD-2 *or the* THRESHOLD-1 *protocol.*

3. Analysis of the Threshold-2 protocol

In this section, we analyse the dimension-exchange protocol THRESHOLD-2 introduced in Section 2. We first show that under this protocol, a distribution always converges to a stable distribution.

Lemma 3. *For an arbitrary initial distribution on a graph G, the dimension-exchange algorithm under the* THRESHOLD-2 *protocol will determine a stable distribution.*

Proof. By Observation 2, $\sum_v \text{load}(v)^2$ is non-increasing with time under the THRESHOLD-2 protocol. A token is moved across an active edge under the THRESHOLD-2 protocol if and only if the discrepancy of the edge is at least two. In this case, $\sum_v \text{load}(v)^2$ decreases by at least two. Since $\sum_v \text{load}(v)^2$ is bounded from below (for a fixed total number of tokens), the number of moves is finite. Therefore, there is some time t after which every active edge has discrepancy at most one. After t there is no movement of tokens, and thus the distribution is stable at t. □

We now characterise stable distributions under the THRESHOLD-2 protocol.

Lemma 4. *A distribution on a graph G is stable under the* THRESHOLD-2 *protocol if and only if every edge of G has discrepancy at most one.*

Proof. ($\Leftarrow$) Observe that if every edge has discrepancy at most one, then during the course of one round there is no movement of tokens, and thus the distribution is stable.

($\Rightarrow$) Suppose that at time t_0 the distribution is stable. By Lemma 2, every active edge after t_0 has discrepancy at most one. Therefore, there is no movement of tokens. In the round starting at t_0 every edge becomes active. Hence, every edge has discrepancy at most one. □

This enables us to prove the main result of this section.

Theorem 5. *Let G be a connected graph with diameter d. Given an arbitrary initial distribution of tokens on G, the* THRESHOLD-2 *protocol will determine a distribution on G with discrepancy at most d.*

Proof. By Lemma 3, the THRESHOLD-2 protocol will determine a stable distribution. Hence the maximum stable discrepancy is an upper bound on the discrepancy of the final distribution.

We now show that the maximum stable discrepancy of G under the THRESHOLD-2 protocol is at most d. Suppose there is a stable distribution on G, and that P is a shortest path from a node with minimum load to a node with maximum load. P has at most d edges and, by Lemma 4, the discrepancy of edges on P is at most one. Hence the difference between the loads of the end-nodes of P is at most d. Hence the global discrepancy is at most d.

We now show that there exists a stable distribution on G with discrepancy d. Let P be a path in G with d edges, and let v be an end-node of P. Set the load of every node w of G to be the graph-theoretic distance from w to v. This distribution has discrepancy d and the discrepancy of every edge is at most one. By Lemma 4, the distribution is therefore stable under the THRESHOLD-2 protocol.

Hence the maximum stable discrepancy of G is d, and therefore the THRESHOLD-2 protocol will determine a distribution with discrepancy at most d. □

4. 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. The THRESHOLD-1 protocol differs from THRESHOLD-2 in that a token is moved across an edge with discrepancy one. In this case, the discrepancy of the edge and $\sum_v \text{load}_t(v)^2$ do not change. Hence the analysis introduced in Section 3 for the THRESHOLD-2 protocol is not applicable to THRESHOLD-1. The following subsection introduces the notion of an observer tour which is subsequently used in the analysis of the THRESHOLD-1 protocol.

4.1. The observer tour

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.

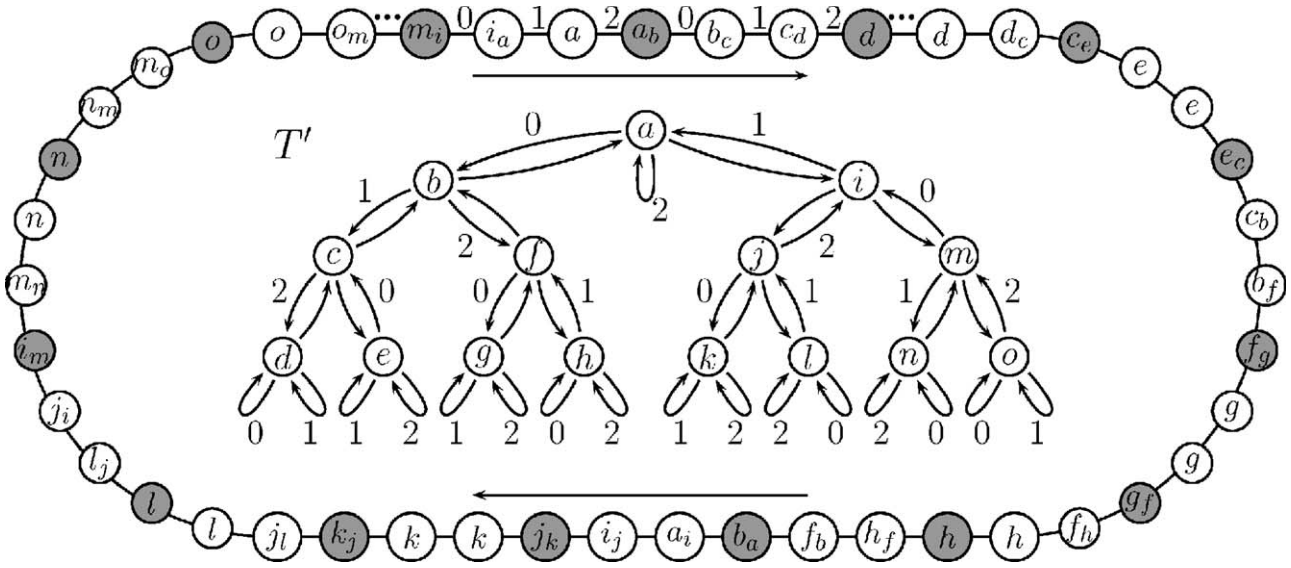

Fig. 1. Observer tour of the complete binary tree of height 3.

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 Fig. 1.

The following lemma enables us to analyse the THRESHOLD-2 protocol for trees.

Lemma 6. *For a tree T, the observer tour defines an Eulerian tour of T'.*

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 $\overrightarrow{vw_c}$, $0 \leqslant c \leqslant \chi - 1$, 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 4. *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

$$\mathrm{gap}_c(v,m) = (j - i) \bmod n.$$

Since $1 \leqslant \mathrm{gap}_c(v,w) \leqslant n-1$, we call an integer $p \in \{1, 2, \ldots, n-1\}$ a *gap* of T. Clearly $\mathrm{gap}_c(v,m) + \mathrm{gap}_c(w,v) = n$. It follows from the definitions that for all nodes v, w, x of T,

$$\mathrm{gap}_c(v,m) = (\mathrm{gap}_c(v,x) + \mathrm{gap}_c(x,w)) \bmod n. \tag{3}$$

In the tree shown in Fig. 1, the 0-ordering is $(a,d,c,e,f,g,h,b,j,k,l,i,n,o,m)$, the 1-ordering is $(b,d,e,c,g,f,h,a,k,j,l,m,n,o,i)$, and the 2-ordering is $(c,d,e,b,g,h,f,i,k,l,j,n,m,o,a)$. The gaps between the nodes a and b are $\mathrm{gap}_0(a,b) = 7$, $\mathrm{gap}_0(b,a) = 8$, $\mathrm{gap}_1(a,b) = 8$, $\mathrm{gap}_1(b,a) = 7$, $\mathrm{gap}_2(a,b) = 4$ and $\mathrm{gap}_2(b,a) = 11$.

4.2. Observers and phases

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 $\mathrm{load}(x) = 1$ and $\mathrm{load}(y) = 0$ then one token is sent from x to y. Hence $\mathrm{load}(x)$ becomes zero and $\mathrm{load}(y)$ 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 6, 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 parallel steps. A phase commencing at time $t \equiv c \bmod \chi$ is called a c-phase.

For our purposes it shall suffice to consider disjoint c-phases for some fixed colour c.

In order to analyse the effects of the THRESHOLD-1 protocol on the circulation of tokens, it will be convenient to view tokens from 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. At each time step, each observer inspects the load of its current node; it is the sequence of load values encountered by an observer that we wish to analyse. We formalise these notions as follows.

Definition 4. Consider a phase of the dimension-exchange algorithm on a tree T starting at time $t_0 \equiv c \bmod \chi$. For each node v of T the *observer* of v at time t, $t_0 \leqslant t < t_0 + \chi \cdot n$, denoted by $\mathrm{obs}_t(v)$, is the node w with $\mathrm{gap}_c(v, m) = t$. We say $\mathrm{obs}(v)$ *is at* w at time t if $\mathrm{obs}_t(v) = w$. (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 $\mathrm{globalMax}_{t_0}(T)$ ($\mathrm{globalMin}_{t_0}(T)$); that is, the maximum (minimum) load at the *start* of the phase.

Suppose that $\mathrm{obs}(v)$ is a maximum (respectively, minimum) observer, and at some time point in a phase, $\mathrm{obs}(v)$ is at a node x and the edge xy is active. If the discrepancy of xy is at most one, then the load of $\mathrm{obs}(v)$ is unchanged as $\mathrm{obs}(v)$ moves to y, as shown in Fig. 2(a) and (b). Otherwise, the discrepancy of xy is at least two, and $\mathrm{obs}(v)$ will no longer be a maximum (minimum) observer after it moves to y, as shown in Fig. 2(c).

We therefore have the following observation.

Observation 5. *Let* $\mathrm{obs}(v)$ *be at node* x *and the edge* xy *be active. Then under the action of the* THRESHOLD-1 *protocol, the load of* $\mathrm{obs}(v)$ *increases/decreases if and only if the edge discrepancy* $\Delta(xy) \geqslant 2$ *and* x *is more lightly/heavily loaded.*

Fig. 3 provides an example of a stable token distribution on the complete binary tree of height two. The colour of each edge and the load of each node after each parallel step is indicated. There are two maximum observers and one minimum observer, each of which remain maximum or minimum observers throughout the phase.

Lemma 7. *Consider a phase of* THRESHOLD-1 *on a tree* T *starting at time* t_0 *and ending at time* t_1. *If* $\mathrm{globalMin}_{t_0}(T) = \mathrm{globalMin}_{t_1}(T)$, *then all nodes* v *with* $\mathrm{load}_{t_1}(v) = \mathrm{globalMin}_{t_1}(T)$ *had* $\mathrm{load}_{t_0}(v) = \mathrm{globalMin}_{t_0}(T)$. *Similarly, if* $\mathrm{globalMax}_{t_0}(T) = \mathrm{globalMax}_{t_1}(T)$, *then all nodes* v *with* $\mathrm{load}_{t_1}(v) = \mathrm{globalMax}_{t_1}(T)$ *had* $\mathrm{load}_{t_0}(v) = \mathrm{globalMax}_{t_0}(T)$.

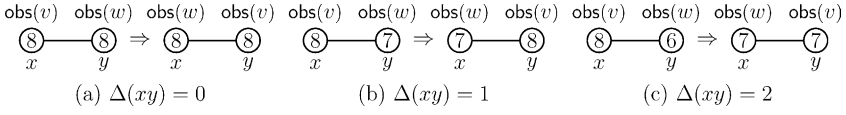

Fig. 2. Movement of an observer under THRESHOLD-1.

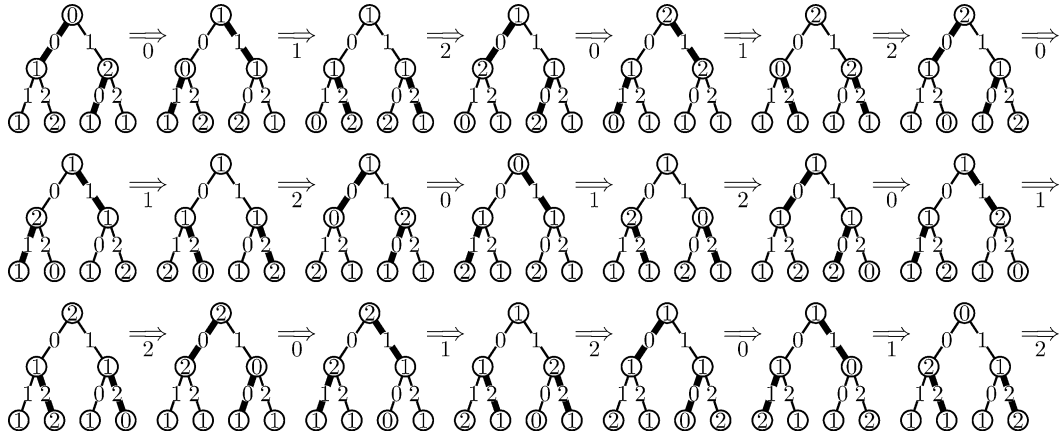

Fig. 3. Example of one phase of THRESHOLD-1.

Proof. We shall prove the result for the minimum load. The proof for the maximum load is analogous. Suppose on the contrary that there exists a node v with $\text{load}_{t_0}(v) > \text{globalMin}_{t_0}(T)$ and $\text{load}_{t_1}(v) = \text{globalMin}_{t_1}(T) = \text{globalMin}_{t_0}(T)$. At some time during the phase the load of obs(v) has decreased to $\text{globalMin}_{t_0}(T)$. Let x and y be the nodes such that by obs(v) moving from x to y the load of obs(v) first decreases from $\text{globalMin}_{t_0}(T)+1$ to $\text{globalMin}_{t_0}(T)$. By Observation 5, for the load of obs(v) to decrease we must have that $\Delta(xy) \geqslant 2$ and $\text{load}(x) = \text{globalMin}_{t_0}(T)+1 \geqslant \text{load}(y)+2$. Hence $\text{load}(y) \leqslant \text{globalMin}_{t_0}(T)-1$. However this contradicts Observation 3, which asserts that the minimum load is on-increasing with time. □

We now show that the dimension-exchange algorithm using the THRESHOLD-1 protocol converges to a stable distribution.

Lemma 8. *For an arbitrary initial distribution on a tree T, under the* THRESHOLD-1 *protocol, the dimension-exchange algorithm will determine a stable distribution.*

Proof. Suppose the active edge vw has discrepancy at least two. In this case, the THRESHOLD-1 protocol will always move a token, and as a result $\sum_v \text{load}(v)^2$ decreases by at least two. Since $\sum_v \text{load}(v)^2$ is bounded below (for a fixed total number of tokens), and since under THRESHOLD-1 $\sum_v \text{load}(v)^2$ is non-increasing, the number of moves across edges with discrepancy at least two is finite. Thus there is some time t after which every active edge has discrepancy at most one. During the phase starting at t, the load of every observer will not change; thus at the completion of the phase each node has the same load as at the start. Therefore the distribution is stable. □

As a result of Lemma 8, the maximum stable discrepancy provides an upper bound on the final discrepancy of a given distribution under the THRESHOLD-1 protocol.

4.3. Maximum stable discrepancy

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 $t \equiv c \bmod \chi$. In Lemma 9 below we show that if $\text{gap}_c(v,m) = |T(x,y)|$ for some pair of nodes v, w and some edge xy, then the node-discrepancy $\Delta_t(v,w) \leqslant 1$. We therefore define the *stable gaps for discrepancy 1* as follows:

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

Since $|T(x,y)| + |T(y,x)| = n$, if $p \in \text{SG}_1(T)$ then $n - p \in \text{SG}_1(T)$. For $T_{2,2}$, the complete binary tree of height two (see Fig. 3), $\text{SG}_1(T_{2,2}) = \{1,3,4,6\}$, and for $T_{3,2}$, the complete binary tree of height three (see Fig. 1), $\text{SG}_1(T_{3,2}) = \{1,3,7,8,12,14\}$.

We now show that the stable gaps for discrepancy 1 determine which observers meet at an active edge during a phase.

Lemma 9. *Under the action of the* THRESHOLD-1 *protocol on a tree T, two observers* obs(v) *and* obs(w) *in a particular c-phase are at end-nodes of a common active edge during this phase if and only if* $\text{gap}_c(v,m) \in \text{SG}_1(T)$.

Proof. ($\Leftarrow$) Suppose $\text{gap}_c(v,m) \in \text{SG}_1(T)$. Then, by the definition of $\text{SG}_1(T)$, there is some edge xy in T with $\text{gap}_c(v,m) = |T(x,y)|$. Throughout the phase, the number of arcs on the observer tour from obs(v) to obs(w) is $\chi \cdot |T(x,y)|$. By Observation 4, the number of arcs from y_x to x_y on the observer tour is $\chi \cdot |T(x,y)|$. Hence, as illustrated in Fig. 4, when obs(v) is at y_x, obs(w) will be at x_y. The edge on the observer tour immediately ahead of an observer is always active. Hence xy is active at this time as required.

($\Rightarrow$) Now suppose that obs(v) and obs(w) are at nodes y and x, respectively at some time point during the phase, and that xy is an active edge. Since the number of edges from y_x to x_y on the observer tour is $\chi \cdot |T(x,y)|$, and the number of edges on the observer tour from one observer to another is constant during a phase, it follows that $\text{gap}_c(v,m) = |T(x,y)|$, and thus $\text{gap}_c(v,m) \in \text{SG}_1(T)$. □

In a stable distribution, whenever two observers meet at an active edge, their discrepancy must be at most one. Since Lemma 9 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 10. *A distribution on a tree T is stable under the* THRESHOLD-1 *protocol at some time $t \equiv c \bmod \chi$ if and*

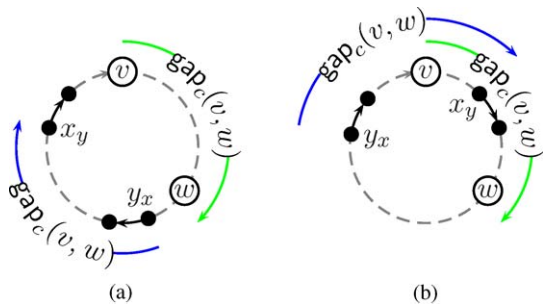

Fig. 4. Relative positions of v, w, and the edge xy in the c-ordering starting at v.

only if every pair v, w of nodes of T with $\text{gap}_c(v,m) \in \text{SG}_1(T)$ has node-discrepancy $\Delta_t(v,w) \leqslant 1$.

Proof. ($\Leftarrow$) If for each pair v, w of nodes of T with $\text{gap}_c(v,m) \in \text{SG}_1(T)$ we have $\Delta_t(v,w) \leqslant 1$, then by Lemma 9, every active edge during the c-phase starting at t has discrepancy at most one. Hence the load of every observer does not change, and at the completion of this phase each node has the same load as at the start. Therefore the distribution is stable.

($\Rightarrow$) Consider a stable distribution on T at time t under the THRESHOLD-1 protocol. Suppose on the contrary, that $\text{gap}_c(v,m) \in \text{SG}_1(T)$ for some pair of node v, w of T, and $\Delta_t(v,w) \geqslant 2$. By Lemma 2, the discrepancy of every active edge after t is at most one. Hence the load of obs(v) and obs(w) will be unchanged after t. By Lemma 9, obs(v) and obs(w) will be at end-nodes of an active edge xy at some time t' during the c-phase starting at t. Therefore $1 \geqslant \Delta_{t'}(xy) = \Delta_{t'}(v,w) \geqslant 2$, which is a contradiction as required. □

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

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

Note that $\text{SG}_i(T)$ is not defined with respect to a particular edge-colouring of T, and for each $i \geqslant 2$,

$$\text{SG}_i(T) = \text{SG}_{i-1}(T) \cup \{(p+q) \bmod n : p \in \text{SG}_{i-1}(T), q \in \text{SG}_1(T)\}. \quad (4)$$

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

$$\text{stability}(p) = \min\{i \geqslant 1 : p \in \text{SG}_i(T)\}.$$

For each gap p, if $p \in \text{SG}_i(T)$ then $n - p \in \text{SG}_i(T)$, and hence $\text{stability}(p) = \text{stability}(n-p)$. Lemma 10 provided our first characterisation of stable distributions under the THRESHOLD-1 protocol. We now provide a second characterisation of stable distributions under the THRESHOLD-1 protocol in terms of the stability of gaps.

Theorem 11. *Let T be a tree whose edges are coloured $0, 1, \ldots, \chi - 1$. Under the* THRESHOLD-1 *protocol, a distribution on T is stable at time $t \equiv c \bmod \chi$ if and only if for all pairs of nodes v, w of T, the node-discrepancy $\Delta_t(v,w) \leqslant \text{stability}(\text{gap}_c(v,m))$.*

Proof. ($\Leftarrow$) Suppose that for all pairs of nodes v, w of T the node-discrepancy $\Delta_t(v,w) \leqslant \text{stability}(\text{gap}_c(v,m))$. Then for all pairs of nodes v, w of T with $\text{gap}_c(v,m) \in \text{SG}_1(T)$, the node-discrepancy $\Delta_t(v,w) \leqslant 1$. By Lemma 10, the distribution is stable.

($\Rightarrow$) 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 $t \equiv c \bmod \chi$ under the THRESHOLD-1 *protocol, then for all pairs of nodes v, w of T with $\text{stability}(\text{gap}_c(v,m)) = i$, the node-discrepancy $\Delta_t(v,w) \leqslant i$.*

The basis of the induction with $i = 1$ is the 'only if' assertion in Lemma 10. Let $i \geqslant 2$, 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 $t \equiv c \bmod \chi$ such that for some nodes v and w with stability $(\text{gap}_c(v,m)) = i$, the node-discrepancy $\Delta_t(v,w) \geqslant i+1$. Thus $\text{gap}_c(v,m) \in \text{SG}_i(T) \backslash \text{SG}_{i-1}(T)$, and hence

$$\text{gap}_c(v,m) = \left(\sum_{j=1}^{i} p_j \right) \bmod n, \quad (5)$$

with $p_j \in \text{SG}_1(T)$. Let x be the node with $\text{gap}_c(v,x) = p_i$. By (3), $\text{gap}_c(v,x) + \text{gap}_c(x,w) = \text{gap}_c(v,m)$. Hence $\text{gap}_c(x,w) = \text{gap}_c(v,m) - p_i$, and by (5),

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

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

The characterisation of stable distributions in Theorem 11 can be used to determine a stable distribution on a tree T with maximum discrepancy. For an n-node tree T, we define

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

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

Theorem 12. *The maximum stable discrepancy of a tree T under the* THRESHOLD-1 *protocol is* $\text{MSD}(T)$.

Proof. By Theorem 11, for every edge-colouring of T with colours $0, 1, \ldots, \chi - 1$, and for each colour $c \in \{0, 1, \ldots, \chi - 1\}$, in a stable distribution on T at time $t \equiv c \bmod \chi$, for all pairs of nodes v, w of T, the node-discrepancy $\Delta_t(v,w) \leqslant \text{stability}(\text{gap}_c(v,m))$. For every gap $p \in \{1, 2, \ldots, n-1\}$, there exist pairs of nodes v, w with $\text{gap}_c(v,m) = p$. Therefore the global discrepancy is at most the maximum of stability(p) taken over all gaps $p \in \{1, 2, \ldots, n-1\}$. Since $\text{SG}_i(T) \subseteq \text{SG}_{i+1}(T)$, this maximum is precisely $\text{MSD}(T)$. Hence there is no stable

distribution on T with greater global discrepancy than $\mathrm{MSD}(T)$.

We now construct, for an arbitrary time t, a distribution on T with discrepancy $\mathrm{MSD}(T)$ which is stable at time t. Let s be an arbitrary node of T. Set $\mathrm{load}(s) \leftarrow 0$, and for every other node v, set $\mathrm{load}(v) \leftarrow \mathrm{stability}(\mathrm{gap}_c(s,v))$ where $t \equiv c \bmod \chi$; that is, $\mathrm{load}(v) = \min\{i \geqslant 1 : \mathrm{gap}_c(s,v) \in \mathrm{SG}_i(T)\}$.

Since $\mathrm{SG}_{i-1}(T) \subseteq \mathrm{SG}_i(T)$ for every $i \geqslant 2$, the discrepancy of this distribution is $\mathrm{MSD}(T)$.

It remains to be shown that the distribution is stable at time t. Consider the action of the THRESHOLD-1 protocol for the c-phase starting at t. Suppose on the contrary, that at some time $t' \geqslant t$ an edge xy with discrepancy at least 2 becomes active. Without loss of generality $\mathrm{load}(x) \geqslant \mathrm{load}(y) + 2$, and t' is the first step of the phase at which such an edge becomes active.

Before step t' in the phase, the discrepancy of every active edge is at most one. Thus, by Observation 5, the load of each observer is unchanged. In particular, if $\mathrm{obs}(v)$ and $\mathrm{obs}(w)$ are at x and y, respectively, then $\mathrm{load}(x) = \min\{i \geqslant 1 : \mathrm{gap}_c sv \in \mathrm{SG}_i(T)\}$ and $\mathrm{load}(y) = \min\{i \geqslant 1 : \mathrm{gap}_c(s,w) \in \mathrm{SG}_i(T)\}$.

Let $j = \mathrm{load}(y)$. Hence $\mathrm{gap}_c(s,w) \in \mathrm{SG}_j(T)$. Since $\mathrm{obs}(v)$ and $\mathrm{obs}(w)$ are end-nodes of an active edge, by Lemma 9, $\mathrm{gap}_c(v,m) \in \mathrm{SG}_1(T)$. By (3), $\mathrm{gap}_c(s,v) = (\mathrm{gap}_c(s,w) + \mathrm{gap}_c(w,v)) \bmod n$, and by (4), $\mathrm{gap}_c(s,v) \in \mathrm{SG}_{j+1}(T)$. Thus $\mathrm{load}(x) \leqslant j+1 = \mathrm{load}(y)+1$, which contradicts our assumption that $\mathrm{load}(x) \geqslant \mathrm{load}(y) + 2$, as required.

Therefore, during the phase there is no active edge with discrepancy at least two. Thus throughout the phase the load of each observer is unchanged, and at the end of the phase, each node has the same load as at the start. Hence this distribution is stable, and therefore the maximum stable discrepancy is $\mathrm{MSD}(T)$. □

Note that for any given n-node tree T, the maximum stable discrepancy of T can be determined by first computing $\mathrm{SG}_1(T)$, and then repeatedly building $\mathrm{SG}_i(T)$ until $\mathrm{SG}_i(T) = \{1,2,\ldots,n-1\}$. Since $\mathrm{SG}_1(T)$ can be computed in linear time by means of a depth-first search of T, and since $\mathrm{SG}_i(T)$ can be computed from $\mathrm{SG}_{i-1}(T)$ according to (4) in $O(n^2)$ time, the maximum stable discrepancy of T can be determined in $O(n^2 \cdot \mathrm{MSD}(T))$ sequential time. The proof of Theorem 12 describes how to compute a distribution on T with maximum discrepancy.

4.4. Bounds for the maximum stable discrepancy

First we establish a linear upper bound on the maximum stable discrepancy of an arbitrary tree under the THRESHOLD-1 protocol.

Lemma 13. *Under the* THRESHOLD-1 *protocol, the maximum stable discrepancy of an n-node tree is at most* $n/2$.

Proof. Observe that, for an arbitrary n-node tree T, if a gap $p \in \{1,2,\ldots,\lfloor n/2 \rfloor\}$ is the sum of at most i terms in $\mathrm{SG}_1(T)$ for some $i \geqslant 1$, then $p \in \mathrm{SG}_i(T)$ and $n - p \in \mathrm{SG}_i(T)$. Since all trees have a node of degree one, $1 \in \mathrm{SG}_1(T)$. If $p \in \{1,2,\ldots,\lfloor n/2 \rfloor\}$ then $p = p \cdot 1 \in \mathrm{SG}_{\lfloor n/2 \rfloor}(T)$ and also $n - p \in \mathrm{SG}_{\lfloor n/2 \rfloor}(T)$. Thus $\mathrm{SG}_{\lfloor n/2 \rfloor}(T) = \{1,2,\ldots,n-1\}$, and by Theorem 12, the maximum stable discrepancy of T is at most $n/2$. □

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

$$C(v) = \max\{|S| : S \text{ is a connected subgraph of } T \backslash \{v\}\}.$$

A centroid is any node v of T for which $C(v)$ is minimum. Kang and Ault [18] have shown that if v is a centroid of T, then $C(v) \leqslant |T|/2$.

Algorithm CONSTRUCT SUBGRAPH (gap p of a tree $T = (V,E)$))
  choose a leaf-node l of T;
  set $\alpha \leftarrow \{l\}$, $\beta \leftarrow V \backslash \{l\}$;
  While $|\alpha| < p$ **do**
    let v be a centroid node of β with $d = \deg(v)$ (see Fig. 5);
    let $\beta_1, \beta_2, \ldots, \beta_d$ be the connected subgraphs of $\beta \backslash \{v\}$ such that
      β_1 contains a node adjacent to a node in α, or
      if v is adjacent to a node in α then $\beta_1 = \emptyset$;
    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**
      set $i \leftarrow \min\left\{ j \geqslant 1 : |\alpha| + 1 + \sum_{k=1}^{j} |\beta_k| > p \right\}$;
      set $\alpha \leftarrow \alpha \cup \{v\} \cup \bigcup_{j=1}^{i-1} \beta_j$;
      set $\beta \leftarrow \beta_i$;
    end-if
  end-while

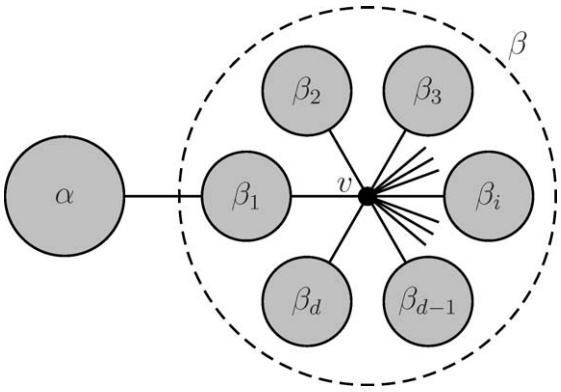

Fig. 5. The subgraphs $\beta_1, \beta_2, \ldots, \beta_d$ associated with a centroid node v of β.

Lemma 14. *Given a gap p of an n-node tree T with maximum degree D, the algorithm* CONSTRUCT SUBGRAPH *determines a connected p-vertex subtree α of T within $\lceil \log_2 n \rceil$ iterations of the while-loop.*

Proof. We now show that at the start of each iteration of the algorithm the following invariants hold: (a) α and β are connected disjoint subtrees, (b) there is one edge between α and β, and (c) $|\alpha| + |\beta| \geqslant p$. Let α' and β' be the new values of α and β after some iteration of the algorithm. In Case 1, $\alpha' = \alpha$ and $\beta' = \beta_1$ are connected subgraphs with one edge between β_1 and α, and $|\alpha'| + |\beta'| = |\alpha| + |\beta_1| > p$. In Case 2, $\alpha' = \alpha \cup \beta_1$ with $|\alpha| + |\beta_1| = p$, thus $|\alpha'| = p$. Therefore following Case 2 the algorithm terminates. In Case 3, α' and β' are connected subgraphs, the edge from v to β_i connects α' and β', and $|\alpha'| + |\beta'| = |\alpha| + 1 + \sum_{k=j}^{i} |\beta_j| > p$.

In Case 1 and 3 of the algorithm, β is replaced by a connected subgraph β_i of $\beta \backslash \{v\}$. By the result in [18] discussed above, $|\beta_i| \leqslant |\beta|/2$. Initially $|\beta| = n - 1$; thus after $\lceil \log_2 n \rceil$ iterations $|\beta| = 0$. Since the algorithm maintains that $|\alpha| + |\beta| \geqslant p$, after $\lceil \log_2 n \rceil$ iterations $|\alpha| \geqslant p$, and thus the algorithm will have terminated. Upon termination, α is a connected subgraph with p vertices. □

Lemma 15. *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 $p \in \mathrm{SG}_M(T)$.*

Proof. Let α be the subgraph produced by CONSTRUCT SUBGRAPH (p). Then $|\alpha| = p$. Let $v_1, v_2, \ldots, v_r$ be the vertices in α incident to an edge whose other endpoint is not in α. Each v_i corresponds to a centroid node chosen in some iteration of the algorithm (when Case 3 is executed), or the node in β_1 adjacent to the centroid node (when Case 1 or Case 2 is executed). In either case each v_i corresponds to a distinct iteration of the algorithm. Hence by Lemma 14, $r \leqslant \lceil \log_2 n \rceil$.

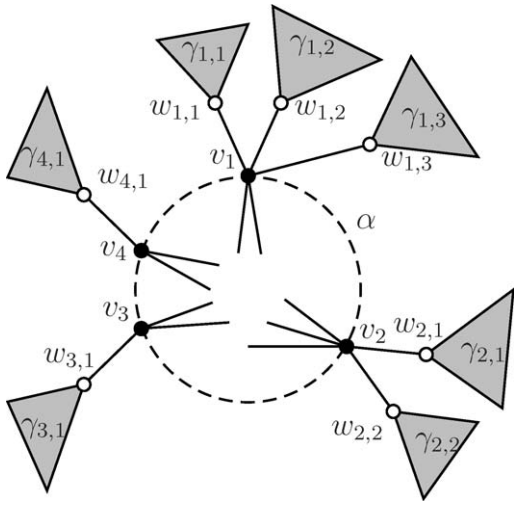

Fig. 6. Vertices v_i and subgraphs of size $\gamma_{i,j}$ associated with a subgraph α.

For each vertex v_i, $1 \leqslant i \leqslant r$, 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 $w_{i,c_i+1}, w_{i,c_i+2}, \ldots, w_{i,\delta_i}$ be the nodes adjacent to v_i which are in α, where $\delta_i = \deg(v_i)$. For $1 \leqslant j \leqslant \delta_i$, let $\gamma_{i,j} = |T(w_{i,j} v_i)|$; see Fig. 6.

Each $\gamma_{i,j} \in \mathrm{SG}_1(T)$ and thus $n - \gamma_{i,j} \in \mathrm{SG}_1(T)$. We can express p as the sum modulo n of terms in $\mathrm{SG}_1(T)$ as follows:

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

Let $M_1 = \sum_{i=1}^{r} c_i$. Then p is the sum modulo n of M_1 terms of $\mathrm{SG}_1(T)$, and hence $p \in \mathrm{SG}_{M_1}(T)$.

We now establish an upper bound on M_1. If v_i is a 'Case 1' or 'Case 2' node then $c_i = 1$. Suppose v_i is a centroid node chosen when Case 3 of the algorithm is executed. If $|\alpha| + |\beta_1| = p - 1$ then α will be replaced by $\alpha \cup \beta_1 \cup \{v\}$ (the algorithm will terminate on the next iteration), and hence $c_i = \delta_i - 1 \leqslant D-1$. If $|\alpha| + |\beta_1| > p - 1$ then at least β_1 and β_2 will be added to α, and hence $c_i \leqslant \delta_i - 2 \leqslant D-2$. Thus $M_1 \leqslant 1 + \sum_{i=1}^{r} (\delta_i - 2) \leqslant 1 + (D-2)r$. Hence $p \in \mathrm{SG}_{1+(D-2)r}(T)$. Since $r \leqslant \lceil \log_2 n \rceil$, $p \in \mathrm{SG}_{1+(D-2)\lceil \log_2 n \rceil}(T)$. This establishes the first part of the result.

We can also express $n - p$ as the sum of terms in $\mathrm{SG}_1(T)$ as follows:

$$\begin{aligned} n - p &= \sum_{i=1}^{r} \sum_{j=1}^{c_i} \gamma_{i,j} \\ &= -r + \sum_{i=1}^{r} \left(1 + \sum_{j=1}^{c_i} \gamma_{i,j} \right) \\ &= -r + \sum_{i=1}^{r} \left(n - \sum_{j=c_i+1}^{\delta_i} \gamma_{i,j} \right) \end{aligned}$$

$$= (n-r) + \sum_{i=1}^{r} \sum_{j=c_i+1}^{\delta_i} (n - \gamma_{i,j}) \pmod{n}.$$

$r = r \cdot 1$ is the sum of r terms of $\mathrm{SG}_1(T)$. Hence $n-r$ is the sum modulo n of at most r terms of $\mathrm{SG}_1(T)$; see Lemma 13. Since each $n - \gamma_{i,j} \in \mathrm{SG}_1(T)$, it follows that $n-p$ is the sum modulo n of at most

$$r + \sum_{i=1}^{r} (\delta_i - c_i) = \sum_{i=1}^{r} (1 + \delta_i - c_i) \leqslant \sum_{i=1}^{r} (D + 1 - c_i) = r(D+1) - M_1$$

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

Since every gap is in $\mathrm{SG}_M(T)$, $\mathrm{MSD}(T) \leqslant M$, and by Theorem 12, the maximum stable discrepancy of T is at most M. Also, by Lemma 13 the maximum stable discrepancy of T is at most $n/2$. By Lemma 8, under the THRESHOLD-1 protocol, the distribution always converges to a stable distribution. We therefore have the following result.

Theorem 16. *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*

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

4.5. Examples

Let S_k be the k-star; that is, the tree with k edges all incident to a single node. Then $\mathrm{SG}_1(S_k) = \{1\}$. It follows that $\mathrm{MSD}(S_k) = \lfloor (k+1)/2 \rfloor = \lfloor n/2 \rfloor$, and hence Lemma 13 is tight for S_k. By Theorem 5, 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 5, the n-node path P_n has maximum stable discrepancy of $n-1$ under the THRESHOLD-2 protocol, and since $\mathrm{SG}_1(P_n) = \{1, 2, \ldots, n-1\}$, Theorem 12 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.

Houle et al. [15], who introduced the THRESHOLD-1 protocol for trees, provided analysis only in the case of the complete binary tree. We now provide two upper bounds on the maximum stable discrepancy of the complete k-ary tree under the THRESHOLD-1 protocol. The first uses an inductive argument and matches the bound in [15] for $k = 2$ (up to an additive constant of 1). The second strengthens the generic result of Theorem 16 to provide a better bound for large values of k.

Lemma 17. *Under the* THRESHOLD-1 *protocol, the maximum stable discrepancy of the complete k-ary tree of height h, $T_{h,k}$ $(k \geqslant 1, h \geqslant 1)$ is at most* $\min\{(k-1)h + 1, (k+2)(h+1)/2\}$.

Proof. We first prove the upper bound of $(k-1)h+1$. The number of nodes in $T_{h,k}$ is $|T_{h,k}| = \frac{k^{h+1}-1}{k-1}$. We proceed by induction on the height h of $T_{h,k}$ (for fixed $k \geqslant 1$) with the following inductive hypothesis. Since each $|T_{j,k}| \in \mathrm{SG}_1(T_{h,k})$, the result will follow.

Every gap p of $T_{h,k}$ is *the sum of at most* $(k-1)h+1$ *terms in* $\{|T_{j,k}| : 0 \leqslant j \leqslant h-1\}$.

The induction basis with $h = 1$ is trivial, since for every gap $p \in \{1, 2, \ldots, |T_{h,k}| - 1\} = \{1, 2, \ldots, k\}$, $p = p \cdot 1$ is the sum of at most k terms in $\{|T_{j,k}| : 0 \leqslant j \leqslant h-1\} = \{1\}$. Let $h \geqslant 2$, and assume that the induction hypothesis holds for $h-1$. For every gap p of $T_{h,k}$,

$$p \leqslant |T_{h,k}| - 1 = \frac{k^{h+1} - 1 - (k-1)}{k-1} = \frac{k(k^h - 1)}{k-1} = k|T_{h-1,k}|.$$

Let q be the quotient and r the remainder when p is divided by $|T_{h-1,k}|$; that is, $p = q|T_{h-1,k}| + r$ with $q \leqslant k$ and $r \leqslant |T_{h-1,k}| - 1$. Suppose $r = 0$. Then $p = q|T_{h-1,k}|$ and $q \leqslant k \leqslant (k-1)h + 1$. Thus the induction hypothesis holds. Suppose $r > 0$. Then $q \leqslant k-1$ and $r \leqslant |T_{h-1,k}| - 1$ is a gap of $T_{h-1,k}$. It follows from the induction hypothesis applied to r with a value of $h-1$ that r is the sum of at most $(k-1)(h-1)+1$ terms of $\{|T_{j,k}| : 0 \leqslant j \leqslant h-2\}$. Thus p is the sum of $q + (k-1)(h-1) + 1 \leqslant (k-1) + (k-1)(h-1) + 1 = (k-1)h + 1$ terms of $\{|T_{j,k}| : 0 \leqslant j \leqslant h-1\}$.

Hence the induction hypothesis is true for all $h \geqslant 1$. It is easily seen that for each j, $0 \leqslant j \leqslant h-1$, $|T_{j,k}| \in \mathrm{SG}_1(T_{h,k})$. Thus each gap p of $T_{h,k}$ is in $\mathrm{SG}_{(k-1)h+1}(T_{h,k})$, and by Theorem 12, the maximum stable discrepancy of $T_{h,k}$ under the THRESHOLD-1 protocol is at most $(k-1)h+1$.

We now prove the upper bound of $(k+2)(h+1)/2$. 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 15 that

the maximum stable discrepancy is at most $(k+2)(h+1)/2$. □

We conjecture that for all $k \geqslant 1$ and $h \geqslant 1$, the maximum stable discrepancy of $T_{h,k}$ under the THRESHOLD-1 protocol is $\lfloor (k-1)h/2 \rfloor$ or $\lfloor (k-1)h/2 \rfloor + 1$. By directly computing $\mathrm{MSD}(T_{h,k})$, this conjecture has been confirmed for the complete binary tree $T_{h,2}$ with $h \leqslant 18$, and for $T_{h,k}$ with $1 \leqslant h \leqslant 6$ and $1 \leqslant k \leqslant 6$.

5. The discrepancy-1 algorithm

In this section, we present algorithm DISCREPANCY-1. This is the first local dimension-exchange algorithm for the token distribution problem on trees that reduces the discrepancy of an arbitrary distribution to at most one. Furthermore, for trees with bounded degree, the rate of convergence is optimal in the worst-case. 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 (for example see Fig. 3). In the THRESHOLD-1 PLUS protocol which follows, 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 $\mathrm{localMax}(v)$, which can be considered a 'local approximation' to the current global maximum. We shall describe later how $\mathrm{localMax}(v)$ is determined. Algorithm DISCREPANCY-1 uses the following dimension-exchange protocol.

Protocol THRESHOLD-1 PLUS (node v, time t)
  if there exists an edge of colour $t \pmod{\chi}$ incident to v **then**
    let vw be this (unique) edge;
    send the value $\mathrm{load}_t(v)$ to w and receive the value $\mathrm{load}_t(w)$ from w;
    if $\mathrm{load}_t(v) \geqslant \mathrm{load}_t(w) + 2$ or
        $(\mathrm{load}_t(v) = \mathrm{load}_t(w) + 1$ and
        $\mathrm{load}_t(v) \neq \mathrm{localMax}(v))$ **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 $\mathrm{localMax}(v)$ is updated so that at the end of the A-phase, for each node v, $\mathrm{localMax}(v)$ is the maximum number of tokens stored at v at any one time during the A-phase. In the B-phase we apply the THRESHOLD-1 PLUS protocol, using the local information gained during the previous A-phase.

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

To assist in the analysis of this algorithm, for a given observer $\mathrm{obs}(v)$ which is maximum (respectively, minimum) at the start of a particular phase, we say that $\mathrm{obs}(v)$ *survives* the phase if it is still a maximum (minimum) observer at the end of the phase. As in Observation 3, it is easily seen that under the THRESHOLD-1PLUS protocol, the maximum (minimum) loads are non-increasing (non-decreasing). Hence the load of a surviving maximum or minimum observer is unchanged throughout the phase.

Let $\mathrm{obs}(v)$ be a surviving maximum or minimum observer in the A-phase, or a surviving minimum observer in the B-phase. The action of the THRESHOLD-1 PLUS protocol for minimum observers is equivalent to that for the THRESHOLD-1 protocol. Thus, by Observation 5, during this phase the discrepancy of an active edge incident to $\mathrm{obs}(v)$ is at most one.

Theorem 18. *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. We first prove that during one cycle of the DISCREPANCY-1 algorithm, the discrepancy will decrease by at least one, assuming that the discrepancy at the start of the cycle is at least two. We proceed by showing 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. Let t_0 be the start of the A-phase, t_1 be the start of the

B-phase, and t_2 be the start of the next cycle. Then $\text{globalMax}_{t_0}(T) - \text{globalMin}_{t_0}(T) \geqslant 2$.

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; that is, $\text{globalMax}_{t_1}(T) = \text{globalMax}_{t_0}(T)$ and $\text{globalMin}_{t_1}(T) = \text{globalMin}_{t_0}(T)$. 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, $\text{localMax}(u) = \text{globalMax}_{t_0}(T)$ for every node u of T.

If at the end of the B-phase the global maximum has decreased; that is, $\text{globalMax}_{t_2}(T) < \text{globalMax}_{t_0}(T)$, then the global discrepancy has decreased by at least one, and we are done. We now assume that $\text{globalMax}_{t_2}(T) = \text{globalMax}_{t_0}(T)$. Thus there exists a node v with $\text{load}_{t_2}(v) = \text{globalMax}_{t_0}(T)$. Now $\text{load}_{t_2}(v) = \text{localMax}(v)$, thus by the definition of Threshold-1Plus, during the B-phase the discrepancy of every active edge incident to v is at most one, and throughout the B-phase $\text{load}(v) = \text{localMax}(v)$. Intuitively, a maximum does not move during the B-phase.

We now prove that the global minimum must increase during the B-phase. Suppose on the contrary, that a minimum observer $\text{obs}(w)$ survives the B-phase; that is, $\text{load}_{t_2}(w) = \text{globalMin}_{t_2}(T) = \text{globalMin}_{t_1}(T)$. Note that the notion of a minimum observer introduced for the Threshold-1 protocol carries over for the Threshold-1 Plus protocol. Now $\text{obs}(w)$ will traverse every node of the tree; in particular, in the step before $\text{obs}(w)$ reaches v, $\text{obs}(w)$ will be at a node u with the edge uv active. In this case $\Delta(uv) = \text{load}(v) - \text{load}(\text{obs}(w)) = \text{globalMax}_{t_0}(T) - \text{globalMin}_{t_0}(T) \geqslant 2$, which contradicts our previous assertion that during the B-phase every active edge incident to v has discrepancy at most one. Since we have assumed that at least one maximum observer survives the B-phase, no minimum observer survives. By Lemma 7, if no minimum observer survives a phase then the minimum load has increased. Hence the minimum load at the end of the B-phase is at least $\text{globalMin}_{t_0}(T) + 1$, and therefore the global discrepancy has decreased by at least one.

Provided that the discrepancy at the beginning of a cycle is at least two, with every cycle the Discrepancy-1 algorithm reduces the discrepancy by at least one. Therefore the number of steps needed 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 in Observation 1 is asymptotically optimal for trees with bounded degree. □

Note that at a node v of T, if $\text{localMax}(v)$ has not changed during two consecutive applications of the Discrepancy-1 algorithm, then the current distribution is stable, and hence the global discrepancy is at most one. In this manner, each node can determine when to terminate the algorithm based purely on local information.

6. Conclusion and open problems

In this paper, we have provided new analyses of two existing dimension-exchange algorithms and introduced a new dimension-exchange algorithm for the token distribution problem on trees. All algorithms are easy to implement and are completely scalable.

The first algorithm reduces the discrepancy of a given distribution on arbitrary graph to at most the diameter of the graph. For the second algorithm, we have presented a number-theoretic method for determining a stable distribution with maximum discrepancy on a given tree. We established a logarithmic upper bound on the maximum stable discrepancy of an arbitrary tree. This protocol has previously been analysed only for the complete binary tree.

Our final algorithm reduces the global discrepancy of an arbitrary distribution on a tree to at most one. To the best of our knowledge, this constitutes the first dimension-exchange algorithm for the token distribution problem on tree-connected architectures that achieves optimal discrepancy.

We now conclude with a number of open problems concerning token distribution problem on trees.

- Can a maximum stable discrepancy of 1 be achieved by an algorithm which stores no local information and has no global knowledge of the network?
- For n-node trees with maximum degree D, the analysis of the randomised algorithm in [10] shows that the discrepancy is reduced to $O(n \log n)$. Is there a randomised strategy for token distribution which performs well on trees?
- For a given graph G we can apply the dimension-exchange algorithm on a fixed spanning tree of G. What properties make a given spanning tree most appropriate?
- Can the methods developed in this paper we extended to the case of Steiner trees and heterogeneous systems?
- Can the work of Xu and Lau [31,32] for optimising the rate at which the discrepancy converges to zero for infinitely-divisible loads, be extended to the case of trees?
- Can the methods developed in this paper be applied in a dynamic setting?

References

[1] I. Ahmad, A. Ghafoor, Semi-distributed load balancing for massively parallel multicomputer systems, IEEE Trans. Software Eng. 17 (10) (1991) 987–1004.

[2] I. Ahmad, A. Ghafoor, G. Fox, Hierarchical scheduling of dynamic parallel computations on hypercube multicomputers, J. Parallel Distrib. Comput. 20 (3) (1994) 317–329.
[3] I. Ahmad, A. Ghafoor, K. Mehrotra, C. Mohan, S. Ranka, Performance modelling of load balancing algorithms using neural networks, Concurrency: Practice Experience 6 (5) (1994) 393–409.
[4] A.Z. Broder, A.M. Frieze, E. Shamir, E. Upfal, Near-perfect token distribution, Random Structures Algorithms 5 (4) (1994) 559–572.
[5] B.S. Chlebus, K. Diks, A. Pelc, Transition-optimal token distribution, Fund. Inform. 32 (3-4) (1997) 313–328.
[6] G. Cybenko, Dynamic load balancing for distributed memory multiprocessors, J. Parallel Distrib. Comput. 7 (1989) 279–301.
[7] C.G. Diderich, M. Gengler, An extended dimension order token distribution algorithm on k-ary d-cubes and its complexity, Internat. J. Found. Comput. Sci. 9 (2) (1998) 213–234.
[8] C.G. Diderich, M. Gengler, S. Ubéda, An efficient algorithm for solving the token distribution problem on k-ary d-cube networks, in: S. Horiguchi, D.F. Hsu, M. Kimura (Eds.), Proceedings of the International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN '94), IEEE, New York, 1994, pp. 75–182.
[9] R. Diekmann, A. Frommer, B. Monien, Efficient schemes for nearest neighbor load balancing, Parallel Comput. 25 (7) (1999) 789–812.
[10] B. Ghosh, F.T. Leighton, B.M. Maggs, S. Muthukrishnan, C.G. Plaxton, R. Rajaraman, A.W. Richa, R.E. Tarjan, D. Zuckerman, Tight analyses of two local load balancing algorithms, SIAM J. Comput. 29 (1) (1999) 29–64.
[11] B. Ghosh, S. Muthukrishnan, Dynamic load balancing by random matchings, J. Comput. System Sci. 53 (3) (1996) 357–370.
[12] C.-C. Han, K.G. Shin, S.K. Yun, On load balancing in multicomputer/distributed systems equipped with circuit or cut-through switching capability, IEEE Trans. Comput. 49 (9) (2000) 947–957.
[13] G. Horton, A multi-level diffusion method for dynamic load balancing, Parallel Comput. 19 (1993) 209–218.
[14] S.H. Hosseini, B. Litow, M. Malkawi, J. McPherson, K. Vairavan, Analysis of graph coloring based distributed load balancing algorithm, J. Parallel Distrib. Comput. 10 (1990) 160–166.
[15] M.E. Houle, E. Tempero, G. Turner, Optimal dimension-exchange token distribution on complete binary trees, Theoret. Comput. Sci. 220 (2) (1999) 363–376.
[16] M.E. Houle, G. Turner, Dimension-exchange token distribution on the mesh and the torus, Parallel Comput. 24 (2) (1998) 247–265.
[17] Y.F. Hu, R.J. Blake, An improved diffusion algorithm for dynamic load balancing, Parallel Comput. 25 (4) (1999) 417–444.
[18] A.N.C. Kang, D.A. Ault, Some properties of a centroid of a free tree, Inform. Process. Lett. 4 (1) (1975) 18–20.
[19] B. Litow, The influence of graph structure on generalized dimension exchange, Inform. Process. Lett. 54 (6) (1995) 347–353.
[20] F. Meyer auf der Heide, B. Oesterdiekhoff, R. Wanka, Strongly adaptive token distribution, Algorithmica 15 (5) (1996) 413–427.
[21] M. Mitzenmacher, How useful is old information?, IEEE Trans. Parallel Distrib. Systems 11 (1) (2000) 6–20.
[22] S. Muthukrishnan, B. Ghosh, M.H. Schultz, First- and second-order diffusive methods for rapid, coarse, distributed load balancing, Theory Comput. Systems 31 (4) (1998) 331–354.
[23] D. Peleg, E. Upfal, The generalized packet routing problem, Theoret. Comput. Sci. 53 (1987) 218–293.
[24] D. Peleg, E. Upfal, The token distribution problem, SIAM J. Comput. 18 (2) (1989) 229–243.
[25] C.G. Plaxton, Load balancing on the hypercube and shuffle-exchange, Technical Report CS-TR-89-1281, Department of Computer Science, Stanford University, August 1989.
[26] A. Radulescu, A.J.C. Van Gemund, Low-cost task scheduling for distributed memory machines, IEEE Trans. Parallel Distrib. Systems 13 (6) (2002) 648–658.
[27] S. Ranka, Y. Won, S. Sahni, Programming a hypercube multicomputer, IEEE Software (1988) 69–77.
[28] G. Turner, H. Schröder, Token distribution on reconfigurable d-dimensional meshes, In: Proceedings of the First IEEE International Conference on Algorithms and Architectures for Parallel Processing, Vol. 1, 1995, pp. 335–344.
[29] V.G. Vizing, On an estimate of the chromatic class of a p-graph, Diskret. Analiz. 3 (1964) 25–30.
[30] C. Xu, B. Monien, R. Lüling, F.C.M. Lau, Nearest-neighbor algorithms for load-balancing in parallel computers, Concurrency: Practice and Experience 7 (7) (1995) 707–736.
[31] C.Z. Xu, F.C.M. Lau, Analysis of the generalized dimension exchange method for dynamic load balancing, J. Parallel Distrib. Comput. 16 (1992) 385–393.
[32] C.Z. Xu, F.C.M. Lau, The generalized dimension exchange method for load balancing in k-ary n-cubes and variants, J. Parallel Distrib. Comput. 24 (1995) 72–85.