J. Parallel Distrib. Comput. 69 (2009) 389–399

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

journal homepage: www.elsevier.com/locate/jpdc

Load-balancing on swapped or OTIS networks

Chenggui Zhao [a,b], Wenjun Xiao [b], Behrooz Parhami [c,*]

[a] Department of Computer Science, Yunnan University of Finance and Economics, Kunming, 650221, China

[b] School of Computer Science and Engineering, South China University of Technology, Guangzhou, 510640, China

[c] Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 931106-9560, USA

ARTICLE INFO

Article history:
Received 12 January 2008
Received in revised form
27 June 2008
Accepted 12 January 2009
Available online 30 January 2009

Keywords:
Diffusion
Dimension exchange
Hierarchical network
Load balancing
Optical transpose interconnect system (OTIS) network
Swapped network

ABSTRACT

Existing local iterative algorithms for load-balancing are ill-suited to many large-scale interconnection networks. The main reasons are complicated Laplace spectrum computations and flow scheduling strategies. Many large-scale networks are modular and/or hierarchically structured, a prime example being the class of swapped or OTIS networks that have received much attention in recent years. We propose a new local scheme, called DED-X, for load-balancing on homogeneous and heterogeneous swapped/OTIS networks. Our scheme needs spectral information only for the much smaller basis or factor graph, which is of size $O(n)$ rather than $O(n^2)$, and it schedules load flow on intragroup and intergroup links separately. We justify the improvements offered by DED-X schemes over traditional X schemes analytically and verify the advantages of our approach, in terms of efficiency and stability, via simulation.

1. Introduction

Load-balancing is the process of redistributing workloads among computational nodes in a parallel or distributed computing environment, when static or a priori distribution fails to achieve near-perfect balance, thus leading to suboptimal efficiency and speedup. In some cases, dynamic, poorly predictable task characteristics bring about a highly uneven load distribution, causing severe performance degradation. Examples abound in cluster and grid computing, which are characterized by a broad collection of dynamically generated loads on processing nodes. Workload equalization is achieved by moving tasks and/or finer chunks of work between nodes and their neighbors via communication links that connect them.

Load-balancing algorithms typically assume that the node workload consists of equally sized items and that the workload is infinitely divisible. The goal of load-balancing is to design scheduling algorithms to migrate load across network links, with each node ideally ending up with a load that matches its capabilities. A node communicates with one neighbor at a time in *dimension exchange* algorithms and with all neighbors simultaneously in *diffusion* algorithms. In parallel processing terminology, these are known as single-port and all-port communication models, respectively. Commonly, load-balancing entails two distinct phases of balance calculation and choice of items to transfer. Balance calculation yields the amount of load that should be migrated between a node and its neighbors to achieve a balanced status. Selection of load items for actual transfer entails a number of criteria relating to workload characteristics. A good load balancing algorithm has a numerically stable iterative process, with low computational complexity and a small flow on communication links for achieving a balanced state.

A number of significant algorithms has been developed for load-balancing on general networks. For homogeneous networks, Cybenko [4] presented a local diffusion load-balancing scheme. Muthukrishnan et al. [13] refer to Cybenko's method as first-order scheme (FOS) and endeavor to speed up its iteration process by using an overrelaxation iterative method in their second-order scheme (SOS). Diekmann et al. [6] developed an iterative algorithm, dubbed the optimal polynomial scheme (OPS), to balance loads among nodes within a finite number of iterations. They also provided a theoretical analysis for their algorithm. Elsässer et al. [7] presented an optimal diffusion scheme, OPT, which balances the node loads in a finite number of iterative steps, once the graph's Laplace spectrum becomes known. They subsequently extended several polynomial diffusion schemes for load-balancing from homogeneous to heterogeneous networks [8], with the resulting "balanced" node loads being proportional to their given weights. The schemes cited above produce l_2-minimal

[*] Corresponding author.
E-mail addresses: zhaochenggui@126.com (C. Zhao), wjxiao@scut.edu.cn (W. Xiao), parhami@ece.ucsb.edu (B. Parhami).

0743-7315/$ – see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jpdc.2009.01.002

balancing flows. In fact, it is the case that all local iterative algorithms lead to a minimal flow, independent of the algorithms and parameters used.

Motivated by load-balancing research in other contexts, Qin [16] designed a data-aware load-balancing strategy to achieve high performance for data-intensive jobs in data grid environments. This was accomplished via a model for estimating the job response time to calculate slowdowns imposed on jobs to balance the load of a data grid in such way that computation and storage resources in each site are simultaneously utilized. Harchol-Balter and Downey [11] proposed a functional form to fit the distribution of lifetimes for Unix processes and derived a preemptive migration load-balancing strategy. They also showed that their policy reduces the mean delay by 35%–50%, compared with other preemptive migration policies.

General diffusion algorithms are ill-suited to load-balancing on large-scale networks, owing to their complicated Laplace spectrum computation. Swapped or OTIS networks constitute a case in point. Optical transpose interconnect system (OTIS) networks, named *swapped* networks by Parhami [15] (who provides a historical review and cites many references to original papers on these networks), are built of n copies of an n-node factor or basis network, and thus have a total of n^2 nodes. Nodes are linked according to the connectivity of the basis network in clusters and via intercluster links to other clusters. The original proposal aimed to realize these intercluster links as optical channels, hence the name "OTIS". Computing the Laplace spectrum of a swapped/OTIS network must take about $O(n^2)$ steps. Interestingly, many algorithms for swapped networks can be based on the respective algorithms/properties of the much smaller basis network. It is thus natural to ask whether spectrum computation for load-balancing can likewise be limited to the basis network in order to make the algorithm much more efficient. Before showing that this simplification can indeed be achieved, we review several studies that deal with load-balancing in large-scale networks.

Several diffusion schemes, described as *alternating direction iteration* (ADI), have been proposed by Elsässer et al. [7] to deal with load equilibrium problems for large and scalable networks. The same research group also introduced *mixed direction iteration* (MDI) in [9] to obtain a smaller flow than ADI, with the same number of iterations. They present ADI-FOS and ADI-OPT, that is, ADI versions of the general diffusion schemes FOS and OPT. When applied to product graphs, the MDI method converges to balanced status faster than the corresponding general diffusion schemes, and the number of iterations is always smaller than the latter. However, these schemes are applicable only to networks modeled as a Cartesian product of two graphs, and thus cannot be used for OTIS architectures.

The diffusion algorithms discussed thus far assume the all-port communication model. For the single-port model, Arndt [1] has constructed the dimension-exchange (DE) algorithms DE-OPT and DE-OPS (which use the same iteration as in OPT and the same iterative polynomial as in OPS diffusion algorithms), but divided each iterative step in both diffusive algorithms into substeps corresponding to edge colors on the underlying graph. DE-OPS and DE-OPT use the most recent information, meaning that a node exchanges its load information with one of its neighbors in each substep. The diffusion matrices of OPT and OPS are replaced with their DE counterparts, and the eigenvalues of the original Laplacian matrix are likewise modified. The convergence speed of DE still depends on the number of distinct eigenvalues of the Laplacian. For product graphs, Arndt [2] developed a new diffusion algorithm ADI-OPS, together with two DE algorithms DE-ADI-OPT and DE-ADI-OPS.

Our aim here is to construct new diffusion algorithms based on general diffusion schemes, such as FOS, SOS, and OPT, so that they can perform load -balancing on OTIS networks with the same level of computational overhead as would be needed for load-balancing on their much smaller basis networks. Accordingly, we propose several hybrid load-balancing schemes and show them to possess a simple iteration process, as well as high efficiency, when applied to a wide array of OTIS networks whose basis networks have regular topologies. Table 1 contains a list of key terms and abbreviations used in concert with other standard graph-theory and parallel-computing terms [10,14].

The rest of this paper is organized as follows. After presenting basic definitions pertaining to load-balancing, diffusion algorithms, and OTIS networks in Section 2, we review the application of several existing general diffusion schemes to homogenous OTIS networks in Section 3. We present our local iterative algorithms for load-balancing on homogeneous swapped/OTIS networks in Section 4, extending the proposed schemes to heterogeneous OTIS networks in Section 5. In Section 6, we analyze the performance of these schemes and present simulation results to show the viability of our approach.

Table 1
Terminology and abbreviations, listed for ready reference.

Term	Meaning or interpretation
ADI	Alternating direction iteration
Adjacency	Matrix representation of a graph
Basis	Component graph from which a swapped/OTIS network is built
Convergence	Speed with which a balanced load status is achieved
CS	Weight assignment with all nodes of weight 1, except one with weight $n+1$
DE	Dimension exchange; communication with one neighbor at a time
DED-X	Three-phase diffusion-exchange-diffusion scheme based on X
Error	Difference between an achieved load distribution and the ideal balanced load
Factor	Same as "basis"
Flow	Extent of workload transfers in a network
FOS	First-order scheme
HOMO	Homogeneous weight assignment: all nodes are given weight 1
Intragroup	Links that connect nodes located in the same basis network
Intergroup	Links that connect nodes located in different basis networks
Laplacian	A particular matrix representation of a graph
Load	Units of work assigned to a particular entity (node, subnetwork, or network)
MDI	Mixed direction iteration
Migration	Transfer of workload among the network nodes
Network	Set of nodes interconnected by links; used interchangeably with "graph"
Norm	Parameter characterizing a diffusion scheme
OPT	Optimal diffusion scheme
OTIS	Optical transpose interconnect system; used interchangeably with "swapped"
OTIS-G	OTIS network with graph G as basis network
OTIS-H_d	OTIS network with a dD hypercube as basis network; generically, OTIS-Cube
OTIS-$M_{k\times m}$	OTIS network with a $k \times m$ mesh as basis network; generically, OTIS-Mesh
PEAK	A highly skewed load distribution where a single node holds the entire load
Quality	Inversely related to extent of flow: the smaller the flow, the higher the quality
RANL	Random load distribution
RANW	Weight assignment, with all node weights being random integers in $[1, 64]$
SEMI	Node assignment in which nodes are given weight 1 or 2, in equal numbers
SOS	Second-order scheme
Stability	Property of an algorithm whose error decreases smoothly and monotonously
Swapped	Used interchangeably with "OTIS"
Weight	Value assigned to a node, reflecting its computational capacity

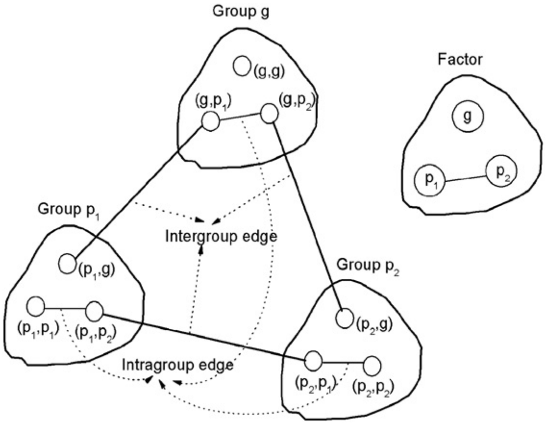


Fig. 1. The structure of swapped/OTIS network.

2. Definitions and background

The swapped/OTIS architecture (see, e.g., [5,15]) derived from a general graph G is denoted as OTIS-G or $S(G)$. A formal definition is given below. Throughout this paper, we use swapped and OTIS networks interchangeably.

Definition 1 (*Swapped/OTIS Graph*). Let $G = (V_G, E_G)$ be an undirected graph. The swapped or OTIS graph associated with G, OTIS-$G = S(G) = (V, E)$, is an undirected graph with the vertex set $V = \{(g, p) \mid g, p \in V_G\}$ and the edge set $E = E_b \cup E_s$, where $E_b = \{((g, p_1), (g, p_2)) \mid g \in V_G, (p_1, p_2) \in E_G\}$ and $E_s = \{((g, p), (p, g)) \mid g, p \in V_G, g \neq p\}$. □

Informally, a swapped/OTIS network, derived from an n-node network G, is composed of n clusters, each of which is internally connected as G. Additionally, node i of cluster j (where $i \neq j$) is connected externally to node j of cluster i.

The graph G is the *factor* or *basis* network of OTIS-G. If G has n nodes, then OTIS-G is composed of n node-disjoint subnetworks G_i, $i = 0, 1, \ldots, n-1$, which constitute the groups or clusters. Each of these groups is isomorphic to G. Denote the vertex set of G_i as $V_i = \{v_{ij} \mid 0 \leq j \leq n-1\}$ and its edge set as $E_i = \{(v_{ik}, v_{il}) \mid (v_k, v_l) \in E_G\}$. The vertex set V of OTIS-G is $V = \cup_{0 \leq i \leq n-1} V_i$. The edge set E of OTIS-G can be partitioned into two subsets: The intragroup or basis edge set E_b, and the intergroup or swap edge set E_s. Clearly, $E_b = \cup_{0 \leq i \leq n-1} E_i$ and $E_s = \{(v_{ij}, v_{ji}) \mid i < j\}$.

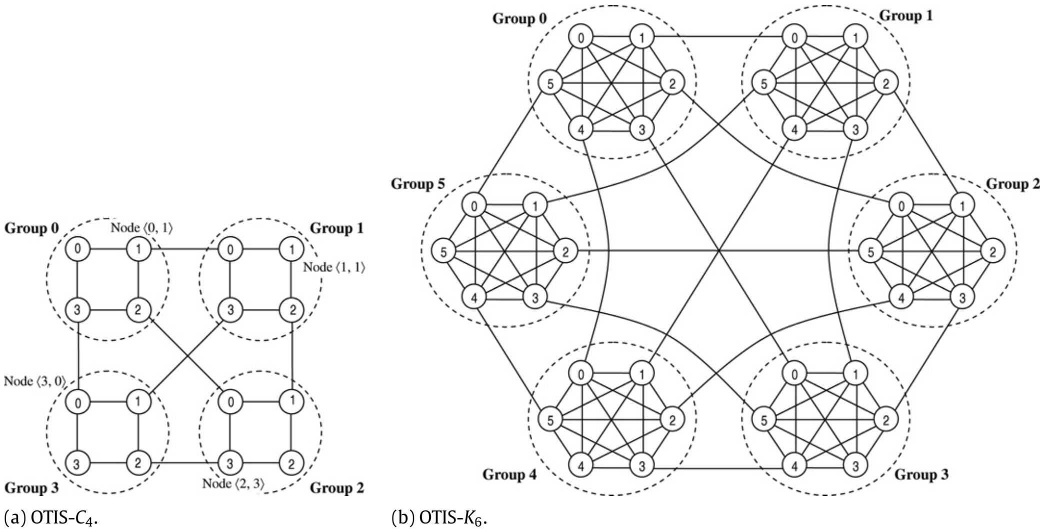
Fig. 1 depicts the structure of a swapped/OTIS network, along with the terminology used to refer to its various parts. Fig. 2(a) contains an example OTIS network formed with the 4-node cycle C_4 as its basis or factor network. The example OTIS network shown in Fig. 2(b) is based on the 6-node complete graph K_6.

Let $w_i = (w_{i1}, w_{i2}, \ldots, w_{in})^{\mathrm{T}}$ and $c_i = (c_{i1}, c_{i2}, \ldots, c_{in})^{\mathrm{T}}$ represent the load and weight on the ith factor G_i of OTIS-G. Similarly, let $w = (w_0, w_1, \ldots, w_{n-1})^{\mathrm{T}}$ and $c = (c_0, c_1, \ldots, c_{n-1})^{\mathrm{T}}$ denote the load and weight on OTIS-G, where the ith components are the load w_i and the weight c_i of G_i. The jth node v_{ij} of the ith basis network has initial load $w_{ij}^0 \geq 0$ and weight $c_{ij} > 0$. Notationally, C and C_i are taken to be diagonal matrices with elements of the vectors c and c_i as their diagonal entries, respectively. That is:

$$C = \mathrm{diag}(c_{01}, \ldots, c_{0n}, c_{11}, \ldots, c_{1n}, \ldots, c_{(n-1)1}, \ldots, c_{(n-1)n})$$
$$C_i = \mathrm{diag}(c_{i1}, c_{i2}, \ldots, c_{in}).$$

Let B_b and B be the node-edge incidence matrices of the basis graph G and OTIS-G respectively; take B_s to be the matrix specifying the incidence of the intergroup edges in E_s to nodes of OTIS-G. Matrices B_b, B_s and B all have in each column exactly two nonzero entries 1 and -1, which represent the nodes incident to the corresponding edge. The signs of these nonzero entries imply directions of the flows produced in the process of load-balancing on the corresponding edges. The Laplacian L of a graph is $L = BB^{\mathrm{T}}$. Let L and L_b be the Laplace matrices of OTIS-G and its basis network G, respectively. Let A_{ij} denote the $n \times n$ matrix with only the ijth entry being 1 and other entries being 0. Let A_s be a matrix with the ijth entry being $A_s(i, j) = A_{ji}$. Then, $L = I_n \otimes (L_b + I_n) - A_s$, where $\otimes$ represents the Knonecker product.

We denote the distinct eigenvalues of L with λ_i ($0 \leq i \leq m$) and those of L_b with λ_i^b ($0 \leq i \leq m_b$), arranged in increasing order. Let $\alpha \in (0, 1)$ be a constant edge weight for OTIS-G and α_b for G. Take $M = I_{n^2} - \alpha L$ and $M_b = I_n - \alpha_b L_b$ to be the corresponding diffusion matrices of polynomial-based diffusion schemes. Then, M and M_b have the eigenvalues $\mu_i = 1 - \alpha\lambda_i$ and $\mu_i^b = 1 - \alpha_b\lambda_i^b$. Denote the second largest eigenvalue of M and M_b according to their absolute values with $\gamma = \max(|\mu_2|, |\mu_m|)$, $\gamma^b = \max(|\mu_2^b|, |\mu_m^b|)$. The workload w^k in step k for polynomial based diffusion schemes can be commonly expressed as a general iteration form $w^k = p_k(M)w^0$. The convergence of this iteration depends on whether the error term $\mathrm{e}^k = w^k - w^l$ tends to zero when k increases, where w^l is the node load vector when the network achieves a balanced status.



(a) OTIS-C_4.

(b) OTIS-K_6.

Fig. 2. Two example OTIS networks, one built of the 4-node cycle C_4 as the basis network and the other based on the 6-node complete graph K_6.

Based on the discussion above, the error term e^k is an indicator of the quality of load-balancing after k steps, and it can be used for comparing distinct algorithms with comparable computational complexities. The iteration error term e^k satisfies (see [6]):

$$\|\mathrm{e}^k\|_2 \le \max\{|p_k(\mu_i)| \cdot \|\mathrm{e}^0\|_2, 2 \le i \le m\}. \tag{1}$$

In particular, the first-order scheme (FOS) satisfies $w^k = Mw^{k-1}$ and yields error results in $\|\mathrm{e}^k\|_2 \le \gamma^k \cdot \|\mathrm{e}^0\|_2$. Since $\gamma(M) = \max(|1 - \alpha\lambda_2(L)|, |1 - \alpha\lambda_m(L)|)$, the minimum of γ is achieved for $1 - \alpha\lambda_2(L) = -1 + \alpha\lambda_m(L)$. Thus, the optimal value of α is $\alpha = 2/(\lambda_2 + \lambda_m)$. Consequently, we have $\gamma = (1-\rho)/(1+\rho)$, with $\rho = \lambda_2(L)/\lambda_m$ constituting the condition number of L.

For OTIS-G, let y^k and x^k be two flow vectors whose entries corresponding to edge e represent the amount of load migrated along e in step k and the total amount of load until step k, respectively. For G, let y_b^k and x_b^k represent the corresponding parameters of G. The directions of the flows are determined by the directions of the edges in the incidence matrix. The flow x is called a balancing flow if and only if $Bx = w^0 - w^l$.

We now proceed to describe several known algorithms, including FOS [4], OPS [6]. and OPT [7]. These all constitute diffusion schemes for load-balancing on networks with general topologies. The FOS scheme can be expressed as a local iterative scheme. It changes the workload vector w^k of nodes and schedules the flow vector x of edges according to $w^k = Mw^{k-1}$, $x^k = x^{k-1} - \alpha B^{\mathrm{T}} w^{k-1}$, $k \ge 1$. The error term e^k of FOS [6] satisfies $\|\mathrm{e}^k\|_2 \le \gamma^k \cdot \|\mathrm{e}^0\|_2$. To improve the relatively slow convergence of FOS, another polynomial-based iterative method, called a second-order scheme (SOS), was devised [12]. The latter is based on the polynomials:

$$p_0(t) = 1, \qquad p_1(t) = t,$$
$$p_k(t) = \beta t p_{k-1}(t) + (1-\beta)p_{k-2}(t) \quad \text{for } k \ge 2.$$

The iterative process of SOS migrates workloads according to $w^1 = Mw^0$, $w^k = \beta M w^{k-1} + (1-\beta)w^{k-1}$, $k \ge 2$. It is known that w^k converges to w^l whenever $\beta \in (0, 2)$, with the fastest convergence occurring for $\beta = 2/(1 + \sqrt{1-\gamma^2})$. Following [6], we denote this optimal value of β as β_0. Then, the error term e^k in the kth iteration satisfies:

$$\|\mathrm{e}^k\|_2 \le (\beta_0 - 1)^{k/2}(1 + k\sqrt{1-\gamma^2})\|\mathrm{e}^0\|_2. \tag{2}$$

After these parameters have been computed, the SOS algorithm can be expressed as a general framework, as suggested in [6].

The optimal scheme OPT [7] has the following simpler iteration process: $2 \le k \le r-1$, $y^{k-1} = (1/\lambda_{k+1})B^{\mathrm{T}}w^{k-1}$, $x^k = x^{k-1} + y^{k-1}$, $w^k = [I - (1/\lambda_{k+1})L]w^{k-1}$.

3. Hybrid diffusion schemes for homogeneous OTIS-networks

For an OTIS-network with n vertices on its basis network, all eigenvalues of an $n^2 \times n^2$ matrix have to be computed before the load-balancing process starts. This is sometimes impractical, motivating us to pursue a hybrid scheme called DED-X, that combines diffusion and dimension exchange, for OTIS networks. Its basic idea is to divide the load-balancing process into three stages of diffusion, exchange, and diffusion. To describe our DED-X approach, let the symbol X denote any known general load-balancing scheme. In the first stage, DED-X iteratively diffuses node loads until the initial load w_i^0 of the ith basis network achieves a balanced status w_i^l locally within the basis network. In this stage, the workload w^k of step k can be expressed as $w^k = (I_n \otimes p_k(M_b))w^0$.

In the second stage, a dimension exchange strategy is applied over all intergroup links. In this stage, basis networks interchange their balanced node load by a way of swapping the load of node (u, v) with that of node (v, u). At the end of this stage, the total load on each basis network is the same. Given the status at the end of the first stage, the load of the entire network after this second stage is given by $w^{l+1} = A_s w^l$.

In the third stage, we proceed with diffusion using the same iterative polynomial-based scheme as in the first stage. However, given that all basis networks have the same initial load vector, we only compute the load migration on one of the basis networks, using the resulting common flow on all other basis networks.

Fig. 3 illustrates the preceding three-stage load balancing process. Note that the key property responsible for DED-X's extreme efficiency is the ability of an OTIS network to disperse the balanced basis network loads uniformly over all clusters, making the load vector in each cluster identical after a single load exchange step via the intercluster (swap) links.

We now proceed to prove that any polynomial-based scheme used in the DED-X framework must force w^k to the average of node loads in the entire network after the completion of DED-X's diffusion-exchange-diffusion process.

Theorem 1. *For any polynomial-based scheme X that takes at most l steps to iteratively balance the load within a basis network, from the initial loads w_i^0 for the ith basis network to the common load $\frac{1}{n}Jw_i^0$, DED-X scheme balances the load w^0 to the common load $\frac{1}{n^2}(J \otimes J)w^0$ in at most $2l + 1$ steps.*

Proof. By the condition of this proposition, $p_l(M_b) = (1/n)J$. Applying the DED-X scheme, we have:

$$\begin{aligned} w^{2l+1} &= [I_n \otimes p_l(M_b)]A_s[I_n \otimes p_l(M_b)]w^0 \\ &= \frac{1}{n^2}(I_n \otimes J)A_s(I_n \otimes J)w^0 = \frac{1}{n^2}(J \otimes J)w^0. \end{aligned}$$

The equation above shows that w^0 will tend to the final balanced load of $\frac{1}{n^2}(J \otimes J)w^0$ within $2l + 1$ steps. □

The DED-X scheme, as applied here to homogeneous systems, will be expressed in the form of a local iterative algorithm in Section 4 (see Fig. 4). Such a DED-X algorithm for heterogeneous networks readily yields DED-X for a homogeneous network as a special case. The performance of these algorithms will be discussed in detail in Sections 5 and 6.

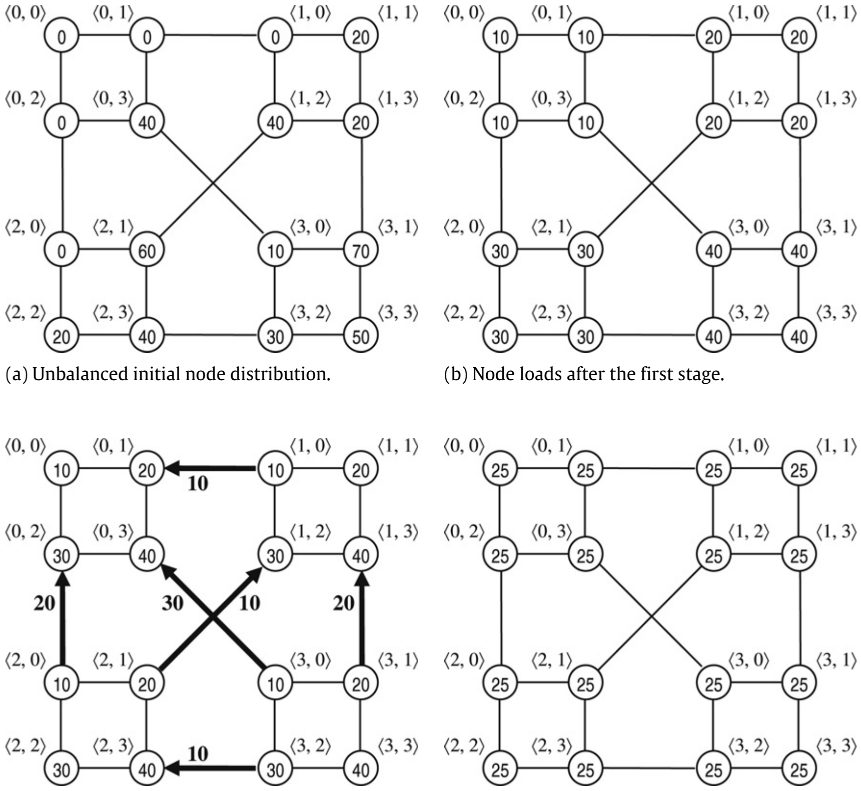
Note that one can select any previously known (e.g., FOS, SOS, OPT) or newly-proposed load-balancing scheme to replace X in DED-X. Thus, DED-X leads to a variety of practical algorithms with diverse attributes.

The flow calculated by DED-X is not minimal in l_2-norm. In the following discussion, λ_i^b, $0 \le i \le m_b$, denotes the Laplacian eigenvalues of the basis graph G and λ_i, $0 \le i \le m$, denotes those of OTIS-G. Let z_i^b be the orthogonal eigenvectors corresponding to λ_i satisfying $\sum_{1 \le i \le n} z_i^b = w_k^0$, where w_k^0 is the part of initial load on the kth basis network. Let z_i be the orthogonal eigenvectors corresponding to λ_i. Take x^g and x^{DED} to represent the flows on OTIS-G links resulting from general diffusion schemes and DED-X schemes, respectively. By Theorem 7 of reference [1], the flow x^g satisfies:

$$x^g = B^{\mathrm{T}} \sum_{i=2}^{m} \frac{1}{\lambda_i} z_i = \begin{bmatrix} (I \otimes B_b^{\mathrm{T}}) \displaystyle\sum_{i=2}^{m} \frac{1}{\lambda_i} z_i \\ B_s^{\mathrm{T}} \displaystyle\sum_{i=2}^{m} \frac{1}{\lambda_i} z_i \end{bmatrix}. \tag{3}$$

If x^{D_1}, x^E, and x^{D_2} represent the flows produced at the first diffusion, exchange, and the second diffusion stages of DED-X schemes, then:

$$x^{\mathrm{DED}} = \begin{bmatrix} x^{D_1} + x^{D_2} \\ x^E \end{bmatrix}. \tag{4}$$



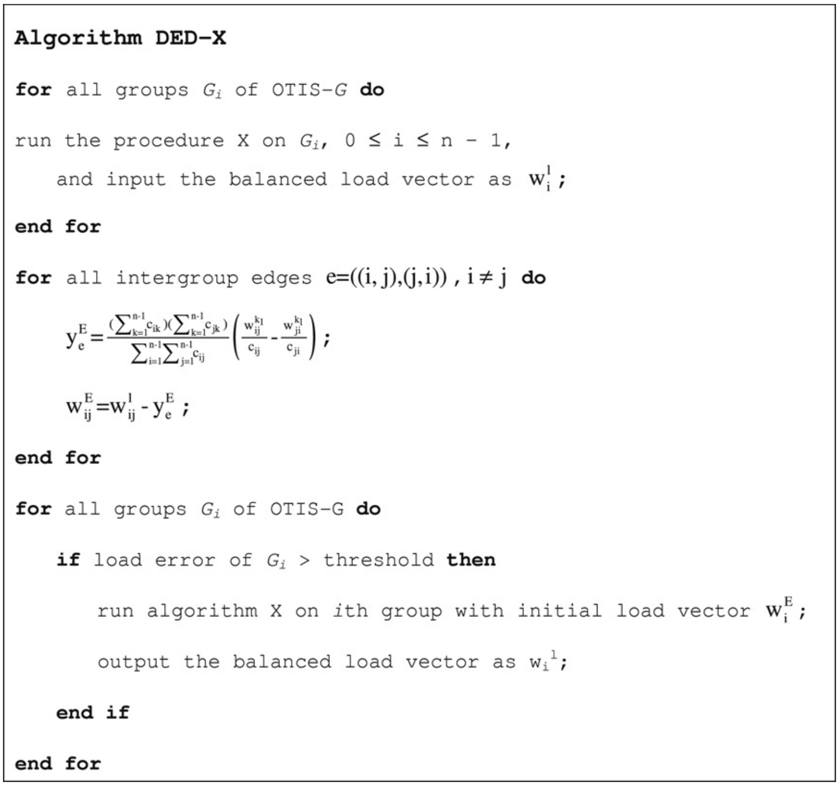
(a) Unbalanced initial node distribution.

(b) Node loads after the first stage.

(c) Node loads after the second stage.

(d) Node loads after the third stage.

Fig. 3. A simple example for the hybrid DED-X schemes, with loads shown at the beginning and after each of the three stages.



```
Algorithm DED-X
for all groups $G_i$ of OTIS-$G$ do
run the procedure X on $G_i$, 0 ≤ i ≤ n - 1,
    and input the balanced load vector as $w_i^1$;
end for
for all intergroup edges $e=((i,j),(j,i))$, $i \neq j$ do
    $y_e^E = \frac{(\sum_{k=1}^{n-1} c_{ik})(\sum_{k=1}^{n-1} c_{jk})}{\sum_{i=1}^{n-1}\sum_{j=1}^{n-1} c_{ij}} \left( \frac{w_{ij}^{k_1}}{c_{ij}} - \frac{w_{ji}^{k_1}}{c_{ji}} \right)$;
    $w_{ij}^E = w_{ij}^1 - y_e^E$;
end for
for all groups $G_i$ of OTIS-G do
    if load error of $G_i$ > threshold then
        run algorithm X on ith group with initial load vector $w_i^E$;
        output the balanced load vector as $w_i^1$;
    end if
end for
```

Fig. 4. The structure of the DED-X algorithm.

Note that we have the following two equalities:

$$x^{D_1} = (I \otimes B_b^{\mathrm{T}}) \begin{bmatrix} \sum_{i=1}^{m_b} \frac{1}{\lambda_i^b} z_{1,i}^b \\ \vdots \\ \sum_{i=1}^{m_b} \frac{1}{\lambda_i^b} z_{n,i}^b \end{bmatrix} = (I \otimes B_b^{\mathrm{T}}) \sum_{i=1}^{n} \sum_{j=1}^{m_b} \frac{1}{\lambda_j^b} (e_i \otimes z_{i,j}^b) \quad (5)$$

$$x^{D_2} = (I \otimes B_b^{\mathrm{T}}) \begin{bmatrix} \sum_{i=1}^{m_b} \frac{1}{\lambda_i^b} z_{*,i}^b \\ \vdots \\ \sum_{i=1}^{m_b} \frac{1}{\lambda_i^b} z_{*,i}^b \end{bmatrix} = (I \otimes B_b^{\mathrm{T}}) \sum_{i=1}^{n} \sum_{j=1}^{m_b} \frac{1}{\lambda_j^b} (e_i \otimes z_{*,j}^b). \quad (6)$$

Substituting $x^E = B_s^{\mathrm{T}} \sum_{1 \le i \le n} e_i \otimes z_{i,1}^b$ as well as the expressions for x^{D_1} and x^{D_2} from Eqs. (5) and (6) into (4), we obtain:

$$x^{\mathrm{DED}} = \begin{bmatrix} (I \otimes B_b^{\mathrm{T}}) \sum_{i=1}^{n} \sum_{j=1}^{m_b} \frac{1}{\lambda_j^b} (e_i \otimes (z_{i,j}^b + z_{*,j}^b)) \\ B_s^{\mathrm{T}} \sum_{i=1}^{n} e_i \otimes z_{i,1}^b \end{bmatrix}. \quad (7)$$

Because $z_{i,j}^b + z_{*,j}^b$ is also an eigenvector of λ_j^b letting $\bar{z}_{i,j}^b = z_{i,j}^b + z_{*,j}^b$, we get:

$$x^{\mathrm{DED}} = \begin{bmatrix} (I \otimes B_b^{\mathrm{T}}) \sum_{i=1}^{n} \sum_{j=1}^{m_b} \frac{1}{\lambda_j^b} (e_i \otimes \bar{z}_{i,j}^b) \\ B_s^{\mathrm{T}} \sum_{i=1}^{n} e_i \otimes z_{i,1}^b \end{bmatrix}. \quad (8)$$

Comparing Eqs. (3) and (8), we can conclude that the flow by DED-X scheme will approach the optimal flow when OTIS-G has relatively complicated eigenvalues with respect to those of G, a condition that is true for most regular basis networks.

4. DED-X schemes for heterogeneous OTIS-networks

The DED-X scheme described thus far cannot achieve the balanced state by only local balancing on basis networks, along with load transposition in the second stage, when the OTIS network is not homogeneous. To generalize the DED-X scheme for use with heterogeneous OTIS networks, it is necessary to revise the load exchange strategy and flow schedule components. In this section, we accomplish this goal by assigning weights to the intergroup links.

With a heterogeneous OTIS network, the X scheme used in the first diffusion phase must be tailored for individual basis networks. Next, we must revise the exchange strategy over intergroup links in the second stage. Using c_{ij} to denote the weight of node (i, j), the weight l_{ij} for the edge $((i, j), (j, i))$, $i \neq j$, linking the basis networks i and j, is assigned as follows:

$$l_{ij} = \frac{\left(\sum_{k=1}^{n-1} c_{ik}\right)\left(\sum_{k=1}^{n-1} c_{jk}\right)}{\sum_{i=1}^{n-1} \sum_{j=1}^{n-1} c_{ij}}. \quad (9)$$

In the third stage, we proceed with diffusion on each of the basis networks by means of the same iterative polynomial as in the first stage, with flow-scheduling on intragroup edges of each basis network.

We can prove that the DED-X scheme must converge from the initial load w^k to the balanced status in the case of heterogeneous networks.

Theorem 2. *For any polynomial based scheme X, if X takes at most k_1 and k_2 steps to redistribute the initial loads w_i^0 and w_i^l of each of basis networks to achieve balanced status, respectively, then the DED-X scheme can lead from the initial load w^0 to the balanced status in at most $k_1 + k_2 + 1$ steps.*

Proof. For any polynomial-based scheme X, and for any $0 \le i \le n - 1$, the scheme X leads the basis network i to local balanced status after the first stage. It follows that:

$$w_{ij}^{k_1} = \frac{c_{ij}}{\sum_{k=1}^{n-1} c_{ik}} \sum_{k=1}^{n-1} w_{ik}^0. \quad (10)$$

In the second stage, based on Eqs. (9) and (10), exchanging loads on intergroup links results in the new loads:

$$\begin{aligned} w_{ij}^{k_1+1} &= w_{ij}^{k_1} - \frac{\left(\sum_{k=1}^{n-1} c_{ik}\right)\left(\sum_{k=1}^{n-1} c_{jk}\right)}{\sum_{i=1}^{n-1} \sum_{j=1}^{n-1} c_{ij}} \left(\frac{w_{ij}^{k_1}}{c_{ij}} - \frac{w_{ji}^{k_1}}{c_{ji}} \right) \\ &= \left(\frac{c_{ij}}{\sum_{k=1}^{n-1} c_{ik}} - \frac{\sum_{k=1}^{n-1} c_{jk}}{\sum_{i=1}^{n-1} \sum_{j=1}^{n-1} c_{ij}} \right) \sum_{k=1}^{n-1} w_{ik}^0 - \frac{\sum_{k=1}^{n-1} c_{ik}}{\sum_{i=1}^{n-1} \sum_{j=1}^{n-1} c_{ij}} \sum_{k=1}^{n-1} w_{jk}^0. \end{aligned} \quad (11)$$

In the third stage, the same iterative polynomial is used, but with different initial loads on the nodes. We thus get, after k_2 additional steps:

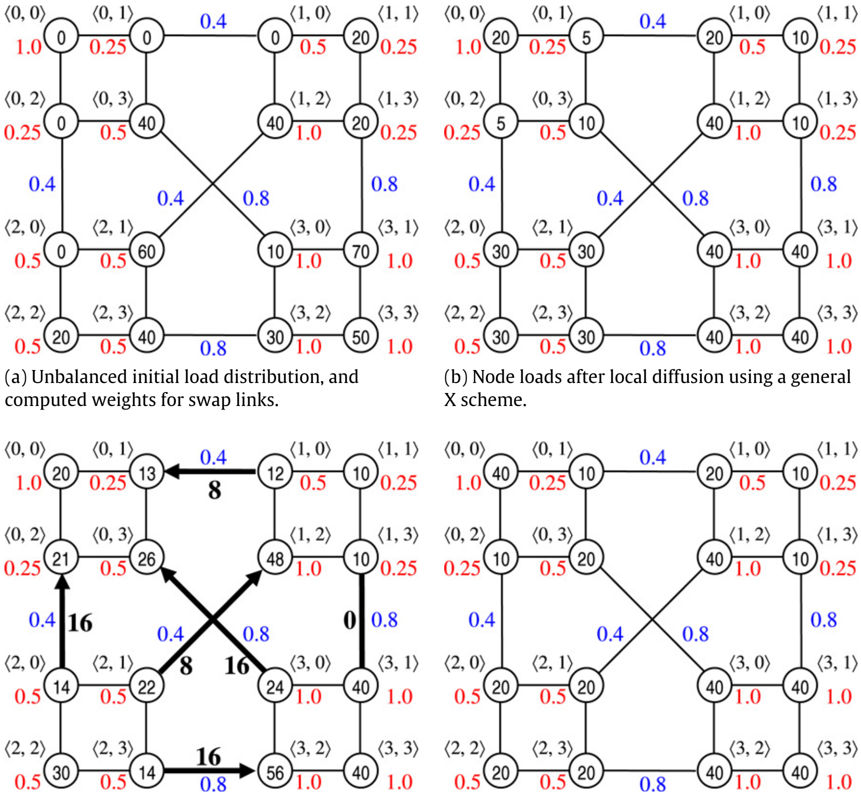
$$\begin{aligned} w_{ij}^{k_1+k_2+1} &= \frac{c_{ij}}{\sum_{k=1}^{n-1} c_{ik}} \sum_{j=1}^{n-1} w_{ij}^{k_1+1} = \frac{c_{ij}}{\sum_{k=1}^{n-1} c_{ik}} \sum_{j=1}^{n-1} \left[\frac{\sum_{k=1}^{n-1} c_{ik}}{\sum_{i=1}^{n-1} \sum_{j=1}^{n-1} c_{ij}} \sum_{k=1}^{n-1} w_{jk}^0 \right] \\ &= \frac{c_{ij}}{\sum_{i=1}^{n-1} \sum_{j=1}^{n-1} c_{ij}} \sum_{j=1}^{n-1} \sum_{k=1}^{n-1} w_{jk}^0 = w_{ij}^l. \end{aligned} \quad (12)$$

Eq. (12) establishes that w^0 tends to w^l within $k_1 + k_2 + 1$ steps. □

The structure of the DED-X algorithm is outlined in Fig. 4 as a local iterative process. As in the case of the homogeneous version of the algorithm, one can select any load-balancing scheme, such as FOS, SOS, or OPT, to replace X. Fig. 5 depicts a simple example of the application of the DED-X algorithm for heterogeneous load balancing. The performance of DED-X schemes will be discussed in detail in Sections 5 and 6.

5. Algorithm analysis

A difference between the DED-X and the X algorithm is that the X scheme can run on any general network, whereas DED-X is specific to OTIS networks. However, when the X algorithm is applied to OTIS-G, Laplacian eigenvalues of the entire OTIS-G graph must be known and iterations are executed on all nodes by flowing loads over all edges synchronously. But with DED-X, only the Laplacian eigenvalues of the basis graph G are necessary and iterations proceed only within groups, separated


⟨0, 0⟩ ⟨0, 1⟩ ⟨1, 0⟩ ⟨1, 1⟩
⟨0, 2⟩ ⟨0, 3⟩ ⟨1, 2⟩ ⟨1, 3⟩
⟨2, 0⟩ ⟨2, 1⟩ ⟨3, 0⟩ ⟨3, 1⟩
⟨2, 2⟩ ⟨2, 3⟩ ⟨3, 2⟩ ⟨3, 3⟩
1.0 0.25 0.5 0.25
0.25 0.5 1.0 0.25
0.5 0.5 1.0 1.0
0.5 0.5 1.0 1.0
0.4 0.4 0.4 0.8 0.8 0.8
(a) 0 0 0 20 / 0 40 40 20 / 0 60 10 70 / 20 40 30 50
(b) 20 5 20 10 / 5 10 40 10 / 30 30 40 40 / 30 30 40 40
(c) 20 13 12 10 / 21 26 48 10 / 14 22 24 40 / 30 14 56 40
8 16 0 8 16 16
(d) 40 10 20 10 / 10 20 40 10 / 20 20 40 40 / 20 20 40 40

(a) Unbalanced initial load distribution, and computed weights for swap links.

(b) Node loads after local diffusion using a general X scheme.

(c) Load exchange on swap links and the resulting loads after exchange.

(d) Node loads after a second local diffusion using the X scheme.

Fig. 5. An example of the DED-X scheme applied to a heterogeneous OTIS network. Node weights in {0.25, 0.5, 1.0} are given immediately below node indices.

by a dimension exchange process over intergroup edges. The most important parameters characterizing the performance of the proposed algorithm include load-balancing accuracy, number of iterative steps, and the amount of flow on communication links. The load-balancing accuracy of DED-X follows from Theorems 1 and 2. Quickness of convergence follows directly from the structure of DED-X, based on the convergence of X. As for the amount of flow, we have analytic results to characterize the extent, but given that the presentation of these results requires the introduction of new notations and extensive derivations, we will report them separately in our future work.

Let $\mathbf{R}^V$ be the set of functions from the vertex set V of graph G to the set $\mathbf{R}$ of real numbers, that is, $\mathbf{R}^V = \{f : V \to \mathbf{R}\}$. We proceed to prove that OTIS-G has a smaller diffusion norm than G, when using a polynomial based diffusion scheme X.

Theorem 3. *Let λ_1^b and λ_1 represent the second smallest Laplace eigenvalue of the basis network G and OTIS-G, respectively. Take $\lambda_{\max}^b$ and $\lambda_{\max}$ to denote their largest eigenvalue, respectively. Then it is true that*

$$\lambda_1 \le \lambda_1^b \tag{13}$$

$$\lambda_{\max}^b + 1 \le \lambda_{\max}. \tag{14}$$

Proof. Let $f \in \mathbf{R}^V$ be a function defined by the eigenvector corresponding to λ_1^b. Then, $f \perp e$, where $e = (1, 1, \dots, 1)^{\mathrm{T}}$. For any $g \in \mathbf{R}^n$, we have $(g \otimes f)^{\mathrm{T}}(e \otimes e) = 0$, so we have $(g \otimes f) \perp e$. Now considering the Rayleigh quotient expression of matrix eigenvalue, and noting that $L_b f = \lambda_1^b f$, we have:

$$\lambda_1 \le \frac{\langle L_s(g \otimes f), g \otimes f\rangle}{\langle g \otimes f, g \otimes f\rangle} = \frac{(g \otimes f)^{\mathrm{T}}(I_n \otimes L_b + I_{n^2} - A_s)(g \otimes f)}{(g \otimes f)^{\mathrm{T}}(g \otimes f)}$$

$$= \lambda_1^b + 1 - \frac{(g^{\mathrm{T}} \otimes f^{\mathrm{T}})A_s(g \otimes f)}{\langle g, g\rangle\langle f, f\rangle}. \tag{15}$$

On the other hand, the following two equations hold:

$$\frac{(g^{\mathrm{T}} \otimes f^{\mathrm{T}})A_s(g \otimes f)}{\langle g, g\rangle\langle f, f\rangle} = \frac{1}{\langle g, g\rangle, \langle f, f\rangle} \sum_{i=1}^{n} g_i f^{\mathrm{T}} \left(\sum_{k=1}^{n} A_{ki} g_k f \right) \tag{16}$$

$$\sum_{i=1}^{n} g_i f^{\mathrm{T}} \left(\sum_{k=1}^{n} A_{ki} g_k f \right) = \sum_{i=1}^{n} g_i f_i f^{\mathrm{T}} g = \langle g, f\rangle^2. \tag{17}$$

Combining Eqs. (15)–(17) we conclude that:

$$\lambda_1 \le \lambda_1^b + 1 - \frac{\langle g, f\rangle^2}{\langle g, g\rangle\langle f, f\rangle}. \tag{18}$$

By choosing g to equal f, we have $\frac{\langle g, f\rangle^2}{\langle g, g\rangle\langle f, f\rangle} = 1$ and $\lambda_1 \le \lambda_1^b$. Eq. (13) is thus satisfied. Similarly, for any $g \in \mathbf{R}^V$, we choose f as an eigenvector of the maximal eigenvalue $\lambda_{\max}^b$ of L_b. Because $g \otimes f \in \mathbf{R}^V$, for $h \in \mathbf{R}^V$ the largest eigenvalue $\lambda_{\max}$ satisfies:

$$\lambda_{\max} = \max_h \frac{\langle L_s h, h\rangle}{\langle h, h\rangle}. \tag{19}$$

Replacing h with $g \otimes f$, we get:

$$\lambda_{\max} \ge \max_g \frac{\langle L_s(g \otimes f), g \otimes f\rangle}{\langle g \otimes f, g \otimes f\rangle}. \tag{20}$$

In a manner similar to the proof of Eq. (13), we can establish that:

$$\frac{\langle L_s(g \otimes f), g \otimes f\rangle}{\langle g \otimes f, g \otimes f\rangle} = \lambda_{\max}^b + 1 - \frac{\langle g, f\rangle^2}{\langle g, g\rangle\langle f, f\rangle}. \tag{21}$$

Table 2
Comparison of diffusion parameters of basis networks and corresponding OTIS networks (G_s stands for OTIS-G).

G	$\lambda_2(G)$	$\lambda_2(G_s)$	$\lambda_m(G)$	$\lambda_m(G_s)$	$\alpha(G)$	$\alpha(G_s)$	$\rho(G)$	$\rho(G_s)$	$\gamma(G)$	$\gamma(G_s)$
H_3	2.0000	0.5857	6.0000	7.4142	0.2500	0.2500	0.3333	0.0790	0.5000	0.8535
$M_{2\times 4}$	0.5857	0.2508	5.4142	6.9319	0.3333	0.2874	0.1081	0.0362	0.8047	0.9301
P_8	0.1522	0.0732	3.8478	5.6542	0.5000	0.3492	0.0396	0.0130	0.9239	0.9744
C_8	0.5858	0.2509	4.0000	5.7491	0.4361	0.3333	0.1464	0.0436	0.7445	0.9164
K_8	8.0000	0.8769	8.0000	10.000	0.1250	0.1839	1.0000	0.0877	0.0000	0.8388

Taken together, Eqs. (20) and (21) yield:

$$\lambda_{\max} \geq \max_{g} \left\{ \lambda^{b}_{\max} + 1 - \frac{\langle g, f\rangle^2}{\langle g, g\rangle\langle f, f\rangle} \right\}. \tag{22}$$

Eq. (14) holds when g is taken to satisfy $\langle g, f\rangle = 0$. □

We now prove that DED-FOS always converges more quickly than FOS.

Theorem 4. *Let γ_b and γ denote the diffusion norms of the basis network G and OTIS-G, respectively. Then $\gamma_b < \gamma$ implies that when applying the DED-FOS scheme and FOS scheme to OTIS-G, DED-FOS will have a smaller upper bound of error than FOS in the kth iteration.*

Proof. Let ρ_b and ρ represent the condition number of the Laplace matrices L_b and L. By using Theorem 3, we find that:

$$\rho = \frac{\lambda_1}{\lambda_{\max}} \leq \frac{\lambda^b_1}{\lambda^b_{\max} + 1} < \frac{\lambda^b_1}{\lambda^b_{\max}} = \rho_b. \tag{23}$$

Therefore:

$$\gamma_b = \frac{1 - \rho_b}{1 + \rho_b} < \frac{1 - \rho}{1 + \rho} = \gamma. \tag{24}$$

Eq. (24) implies that for the OTIS-G network, the iteration error for DED-FOS at the kth step satisfies $\|\mathrm{e}^k_b\|_2 \leq \gamma^k_b$. The corresponding FOS error is $\|\mathrm{e}^k\|_2 \leq \gamma^k$, with $\gamma_b < \gamma$. □

Our next result pertains to the performance of the DED-SOS scheme, showing that it displays better convergence compared with the ordinary SOS scheme for OTIS networks.

Theorem 5. *The upper bounds $b(\mathrm{e}^k_b)$ of iteration error on G and $b(\mathrm{e}^k)$ on OTIS-G utilizing a general SOS scheme satisfies*

$$b(e^k_b) = \tau^{2k-1} b(\mathrm{e}^k) \tag{25}$$

where $\tau < 1$ is a real constant determined from the ratio of diffusion norms γ_b and γ.

Proof. In Eq. (2), let $\sqrt{1 - \gamma_b^2} = \theta$ and $\sqrt{1 - \gamma^2} = \varepsilon$. Then, given that $0 \leq \varepsilon < \theta < 1$, we are led to:

$$\frac{b(e^k_b)}{b(\mathrm{e}^k)} = \left[\frac{(1-\theta)(1+\varepsilon)}{(1+\theta)(1-\varepsilon)} \right]^{k/2} \left(\frac{1 + k\theta}{1 + k\varepsilon} \right). \tag{26}$$

Letting $(1+\varepsilon)/(1+\theta) = \delta$, we obtain $\delta < 1$ and $(1-\theta)/(1-\varepsilon) = \delta\gamma^k_b/\gamma^2$. Now, given that $(1 + k\theta)/(1 + k\varepsilon) < \theta/\varepsilon$ and $k > 0$, we have:

$$\frac{b(e^k_b)}{b(\mathrm{e}^k)} < \delta^{k-1}(\gamma_b/\gamma)^k. \tag{27}$$

Let $\tau = \max(\delta, \gamma_b/\gamma)$. Then, $\tau < 1$, and Eq. (27) becomes Eq. (25).

Theorem 5 implies that, when applied to OTIS-G, DED-SOS will have a smaller error upper bound than SOS in the kth iteration. With regard to the DED-OPT scheme, let d be the diameter of the basis graph. Then, we have the following result.

Theorem 6. *In the worst case, to achieve balanced status on the OTIS network, DED-OPT has a lower bound $d+1$ on the number of iterations required, whereas OPT needs as least $2(d + 1)$ iterations in this case.*

Proof. In the worst case, the OPT scheme requires a number of iterations equal to the number of distinct eigenvalues of the Laplace matrix of the OTIS topology. But the number of iterations required by DED-OPT equals the number of distinct eigenvalues of the basis graph. We know that the Laplacian of a graph always has the same number of distinct eigenvalues as its adjacency matrix; the latter has at least $d + 1$ distinct eigenvalues [3]. Note that a swapped or OTIS network has diameter $2d+1$ when its basis graph has diameter d (see [15]). □

6. Experimental results

We now present some quantitative results based on simulations to help illustrate the properties and advantages of the DED-X scheme in comparison with a general diffusion scheme X. These results are based on some familiar basis networks, including hypercube (H_d), mesh ($M_{k\times m}$), linear array or path (P_n), ring or cycle (C_n) and complete graph (K_n). Each of the OTIS-G networks considered consists of 64 nodes (that is, $n = 2^d = k \times m = 8$). To highlight the impact of the initial load, we experiment with both a highly unbalanced and a randomly distributed initial load, described in the following.

- PEAK: One node has a superload of $100n$, while all others have load 0
- RANL: A total load of $100n$ is randomly distributed among all nodes.

We consider four types of weight assignment with regard to network heterogeneity:

- HOMO: All nodes are given weight 1 (homogeneous)
- SEMI: Nodes are given weight 1 or 2, in equal numbers
- CS: All nodes are of weight 1, except one with weight $n + 1$
- RANW: Node weights are random integers in $[1, 64]$.

Our aim is to compare the new strategies DED-FOS and DED-OPT with the traditional FOS and OPT in speed, flow quality, and numerical stability. We did experiments on mesh and hypercube network with PEAK and RAN initial load distributions. The load-balancing process stops when the error $\|w^k - w^l\|_2$ is less than 0.01.

6.1. Convergence speed

Table 2 presents several diffusion parameters used in X and DED-X schemes, coming from spectral computation of the basis network G and the corresponding OTIS-G. Note that the symbol G_s is used in Table 2 as a shorthand for OTIS-G. All OTIS architectures listed have a lower value for the second smallest eigenvalue λ_2 and condition number ρ of the Laplacian, and larger diffusion norm γ, compared with their corresponding basis networks. These experimental results validate the theoretical derivations of Section 5. That is, our theorems showing that DED-FOS and DED-SOS schemes should converge faster than FOS and SOS,

Table 3
Comparison of the number of the iterations of general scheme X and DED-X (N_{Y} represents the number of iterations of Y).

Network	Initial load	Node weight	N_{FOS}	N_{GDED}-FOS	N_{SOS}	N_{GDED}-SOS	N_{OPT}	N_{GDED}-OPT
OTIS-H_3	PEAK	HOMO	77	38	27	25	15	7
		SEMI	77	38	27	25	15	7
		CS	73	35	26	23	15	7
	RAN	HOMO	76	37	27	25	15	7
		SEMI	76	37	27	25	15	7
		CS	71	38	26	23	15	7
OTIS-$M_{2\times4}$	PEAK	HOMO	165	114	45	38	42	13
		SEMI	165	114	45	38	42	13
		CS	155	105	43	35	42	13
	RAN	HOMO	157	112	45	38	42	13
		SEMI	157	112	45	38	42	13
		CS	144	112	43	35	42	13

Table 4
Comparison of the flows of general OPT and DED-OPT.

Network	Node load	Node weight	OPT flows	DED-OPT flows
OTIS-H_3	PEAK	HOMO	4 599	4636
		SEMI	4 517	4947
		CS	2299	2104
	RAN	HOMO	2 885	3107
		SEMI	2 760	3497
		CS	2 105	1547
OTIS-$M_{2\times4}$	PEAK	HOMO	47 213	6279
		SEMI	53 662	6609
		CS	37 794	2985
	RAN	HOMO	70 986	4971
		SEMI	134 687	5312
		CS	85 719	2075

because they use the same diffusion parameters required for G to implement diffusion on OTIS-G, have been validated.

In Table 3 we present the results of the number of iterations in order to compare the convergence speed of traditional X schemes with those of DED-X schemes, when started on the PEAK and RAN initial load distributions. Our theoretical analysis showed that DED-FOS requires a smaller number of iterations than the traditional FOS scheme. The number of iterations for DED-FOS with PEAK and RAN are nearly identical. DED-OPT behaves better than OPT, requiring nearly half the number of iterations for PEAK and RAN, except in the case of the complete graph K_n.

6.2. Solution quality

It is known (see [6]) that existing polynomial-based schemes all compute the same flow. This suggests that all DED-X schemes have the same flows in the first and third stages. Because the second stage, dimension exchange, is based on the initial state of local-balanced status, DED-X schemes also have the same flows. Consequently, all DED-X schemes compute the same (possibly non-minimal) flow. Taking this into consideration, we only give the flows of DED-OPT on OTIS networks to illustrate the solution quality of DED-X schemes. Table 4 shows that in most cases, the flow of the DED-OPT scheme is smaller than the flow of general OPT. The advantage of DED-OPT over OPT is particularly pronounced in the case of OTIS-$M_{2\times4}$ and OTIS-P_8. Therefore, there is strong incentive for applying DED-X in these cases.

6.3. Numerical stability

As stated previously, the criteria for judging the quality of load-balancing algorithms also include their numerical stability. An iterative algorithm is considered stable if the error e^k in kth step monotonically decreases with respect to the iteration parameter k. Unfortunately, finding a scheme with stable iterative behavior and fewer steps is often not easy. The FOS scheme has good numerical stability, but it is slow. OPT needs a relatively smaller number of iterative steps, but suffers from a possible numerical trap. Figs. 6 and 7 show that the DED-FOS scheme is numerically stable and converges faster than FOS on OTIS architectures.

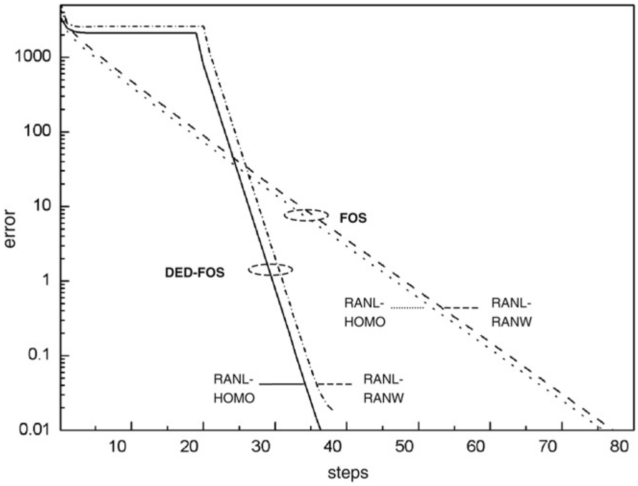


Fig. 6. DED-FOS and FOS on OTIS-H_3 network.

Figs. 6 and 7 also show that DED-FOS is stable in both diffusion stages, with the error steadily decreasing. There is a sudden drop of error arising in the second stage (load exchange), but given the use of optical links in this stage, this may not be a problem.

The best behavior of DED-X can be observed in Figs. 8 and 9, where DED-OPT has a much better stability than general OPT. Relative to OPT, numerical problems are almost never observed in DED-OPT. It is clear that OPT leads to a relatively high peak

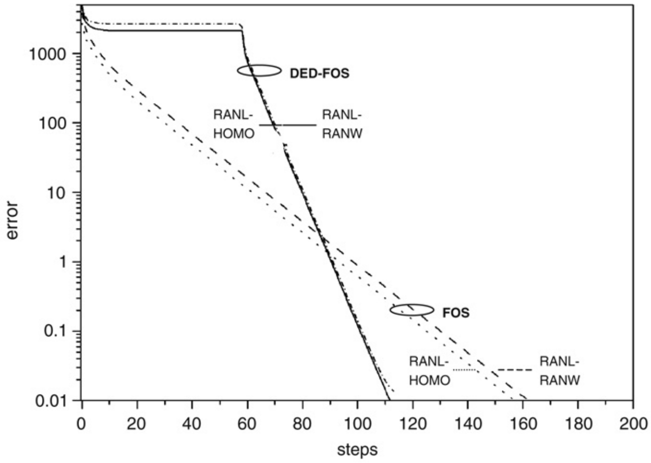


Fig. 7. DED-FOS and FOS on OTIS-$M_{2\times4}$ network.

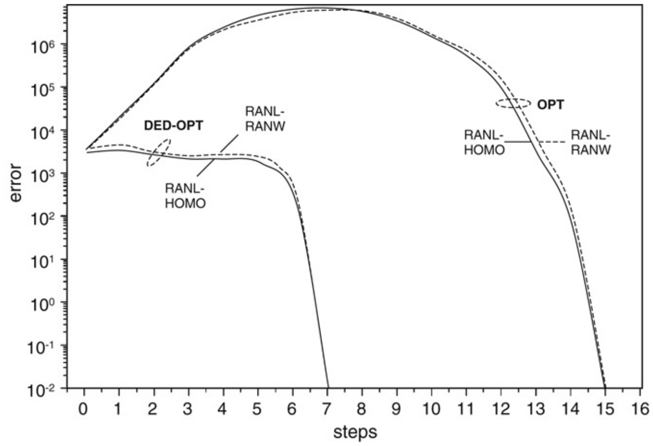


Fig. 8. DED-OPT and OPT on OTIS-H_3 network.

error value, while DED-OPT leads to a significant decrease in iteration error. The abrupt drop to a balanced status in the last iterative step is a common property of convergence for the general OPT scheme [7]. This indicates that a general OPT scheme would possibly fail to load-balance an OTIS architecture with a large basis network, owing to numerical problems and the high error values. In such application domains, DED-OPT might be an appropriate substitute that avoids the stability and error problems.

7. Conclusions

We have presented, theoretically analyzed, and experimentally evaluated a number of algorithms for load-balancing on OTIS networks, demonstrating that the class of diffusion-exchange-diffusion algorithms based on the diffusion scheme X on the basis networks (DED-X) provides highly efficient alternatives to standard algorithms directly adapted to an OTIS network as a whole. These algorithms are useful practically, despite the fact that (like their counterparts to which they are compared) they do not produce optimal flows. The key to the efficiency and greater stability of these algorithms is that they take advantage of the special structure of swapped/OTIS networks to reduce the number of iterations and/or reduce the required communication traffic.

Our work has revealed key properties of load balancing algorithms for the important class of swapped/OTIS networks, in turn leading to opportunities for additional research in this area. We have analyzed the flow generated by DED-X algorithms and will report our results in the near future. A main focus of our ongoing work is considering whether the number of iterations can be further reduced, without creating numerical stability problems or excessive errors. Seeking generalizations to other large-scale networks is also an attractive area for further investigation. While the techniques used here may be inapplicable to other large-scale and hierarchical interconnection schemes, the insights gained from this study may help us identify the appropriate approach in each case.

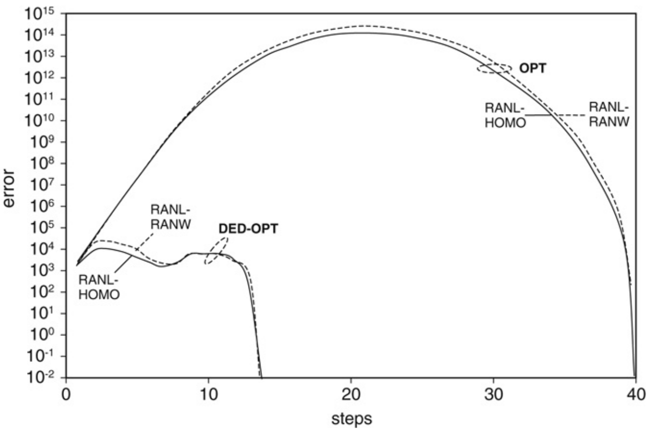


Fig. 9. DED-OPT and OPT on OTIS-$M_{2\times4}$ network.

References

[1] H. Arndt, On finite dimension exchange algorithms, Linear Algebra and its Applications 380 (2004) 73–93.

[2] H. Arndt, Load balancing: Dimension exchange on product graphs, in: Proc. 18th Int'l Parallel and Distributed Processing Symp., 2004, p. 20b.

[3] F.R.K. Chung, Spectral Graph Theory, in: CBMS Regional Conference Series in Mathematics, vol. 92, American Mathematical Society, 1997.

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

[5] K. Day, A.-E. Al-Ayyoub, Topological properties of OTIS-networks, IEEE Transactions on Parallel and Distributed Systems 13 (4) (2002) 359–366.

[6] R. Diekmann, A. Frommer, B. Monien, Efficient schemes for nearest neighbor load balancing, Parallel Computing 25 (7) (1999) 789–812.

[7] R. Elsässer, A. Frommer, B. Monien, R. Preis, Optimal and alternating direction load balancing schemes, in: Proc. 5th Int'l Euro-Par Conf. Parallel Processing, in: LNCS, vol. 1685, Springer, 1999, pp. 280–290.

[8] R. Elsässer, B. Monien, R. Preis, Diffusion schemes for load balancing on heterogeneous networks, Theory of Computing Systems 35 (3) (2002) 305–320.

[9] R. Elsässer, B. Monien, R. Preis, A. Frommer, Optimal diffusion schemes and load balancing on product graphs, Parallel Processing Letters 14 (1) (2004) 61–73.

[10] G. Godsil, Algebraic Graph Theory, Springer, 2001, pp. 279–306.

[11] M. Harchol-Balter, A.B. Downey, Exploiting process lifetime distributions for dynamic load balancing, ACM Transactions on Computer Systems 15 (3) (1997) 253–285.

[12] B. Mohar, Some applications of Laplace eigenvalues of graphs, in: G. Hahn, G. Sabidussi (Eds.), Graph Symmetry: Algebraic Methods and Applications, Kluwer, 1997, pp. 225–275.

[13] S. Muthukrishnan, B. Ghosh, M.H. Schultz, First- and second-order diffusive methods for rapid, coarse, distributed load balancing, Theory of Computing Systems 31 (4) (1998) 331–354.

[14] B. Parhami, Introduction to Parallel Processing: Algorithms and Architectures, Plenum, 1999.

[15] B. Parhami, Swapped interconnection networks: Topological, performance, and robustness attributes, Journal of Parallel and Distributed Computing 65 (11) (2005) 1443–1452.

[16] X. Qin, Design and analysis of a load balancing strategy in data grids, Future Generation Computer Systems 23 (1) (2007) 132–137.

Chenggui Zhao received a B.S. degree in Mathematics and an M.S. degree in Computer Science from Yunnan Normal University, Kunming, in 1999 and 2003, respectively, and the Ph.D. degree in Computer Science from South China University of Technology, Guangzhou, in 2007. He is presently an instructor in the Department of Computer Science, Yunnan University of Finance and Economics, China. His research interests lie mainly in parallel architectures, performance evaluation, and algorithms. He is also interested in P2P computing and computer vision.

Wenjun Xiao received the Ph.D. degree in Mathematics from Sichuan University, People's Republic of China, in 1989. Currently, he is a professor in the School of Computer Science and Engineering, South China University of Technology, Guangzhou, People's Republic of China. His research interests include discrete mathematics, parallel and distributed computing, complex networks, and software architecture. He has published more than 60 papers in international conferences and journals, including *IEEE Transactions on Computers* and *IEEE Transactions on Parallel and Distributed Systems* on these topics since 1985.

Behrooz Parhami received the Ph.D. degree in Computer Science from University of California, Los Angeles, in 1973. Currently, he is a professor in the Department of Electrical and Computer Engineering, University of California, Santa Barbara, USA. His research deals with parallel architectures and algorithms, computer arithmetic, and reliable computing. In his previous position with Sharif University of Technology in Tehran, Iran (1974–1988), he was also involved in the areas of educational planning, curriculum development, standardization efforts, technology transfer, and various editorial responsibilities, including a five-year term as editor of Computer Report, a Persian-language computing periodical. Dr. Parhami's technical publications include more than 250 papers in journals and international conferences, a Persian-language textbook, and an English/Persian glossary of computing terms. Among his latest publications are two graduate-level textbooks on parallel processing (Plenum, 1999) and computer arithmetic (Oxford, 2000), and an introductory textbook on computer architecture (Oxford, 2005). Dr. Parhami is a fellow the IEEE and the IEEE Computer Society, a chartered fellow of the British Computer Society, a member of the ACM, and a distinguished member of the Informatics Society of Iran, for which he served as a founding member and president during 1979–1984. He also served as chairman of the IEEE Iran Section (1977–1986) and received the IEEE Centennial Medal in 1984.