To Improve the Convergence and Parallelism of Gauss-Seidel Routing Algorithm with Finite Element Method for Wireless Sensor Networks

Ren-Song Ko
Department of Computer Science and Information Engineering
National Chung Cheng University, Chia-Yi, Taiwan 621
E-mail: korenson@cs.ccu.edu.tw

***Abstract*—The scalability challenge of many problems in massively-dense wireless sensor networks may be mitigated from a macroscopic perspective. One example is the weak formulation of the load-balancing routing problem, which solution can be used to route information. Hence, a routing algorithm, the distributed Gauss-Seidel iteration (DGSI), was proposed to coordinate sensors to solve the weak formulation iteratively. In this paper, we propose the atomic red-black distributed Gauss-Seidel iteration with finite element method (ARB-DGSI-FEM) to eliminate the early termination problem of DGSI due to the presence of holes, and thus improve the accuracy of numerical solutions. In addition, ARB-DGSI-FEM allows the values of unknowns to be updated in the red-black order to achieve the maximum degree of parallelism and reduce the convergence time. Our simulation results reveal that ARB-DGSI-FEM significantly improves the parallelism without too much sacrifice of accuracy.**

***Index Terms*—wireless sensor networks, load-balancing routing, partial differential equations, finite element method, Gauss-Seidel method**

I. INTRODUCTION

Though advances in hardware miniaturization enable the possibility of a large scale wireless sensor network (WSN) [1], it becomes difficult to keep sensors operating in an efficient manner. For instance, one important consideration in designing an energy-efficient routing algorithm is the load-balancing due to the limited energy capacity and the coverage requirement imposed on a WSN. However, to balance the routing load of all sensors is a global optimization problem and is subject to the scalability problem. Though several load-balancing algorithms have been proposed for WSNs [2], [3], these algorithms determine the next forwarding sensor nodes based on local information, which may make the results less optimal.

The scalability problem may be alleviated by ignoring microscopic details for the massively-dense WSN in which a huge amount of sensors are deployed in a small region. For example, Ko [4] derived the weak formulation of the load-balancing routing problem from a macroscopic perspective,

$$\mathbf{D} = \jmath \nabla \Phi, \tag{1}$$

and

$$\int_A \jmath \nabla \Phi \cdot \nabla \nu \mathrm{d}y\mathrm{d}x = -\int_A \rho \nu \mathrm{d}y\mathrm{d}x. \tag{2}$$

This research was supported by Ministry of Science and Technology of Taiwan, under Grant MOST 106-2221-E-194-027. The authors gratefully acknowledge this support.

Here A and $\rho(x, y)$ respectively represents the region of interest (ROI) and the amount of information generated at $(x, y) \in A$. $\mathbf{D} : A \to \mathbb{R}^2$ is the *routing vector field* in which the direction of $\mathbf{D}(x, y)$ points to the next forwarding node of the sensors at (x, y) and the length $\|\mathbf{D}(x, y)\|$ represents the amount of information transmitted by all sensors at (x, y). In addition, ν is an arbitrary smooth scalar valued function.

The relationship between $\jmath$ and the sensor density ψ may be established once the communication energy consumption is known. For instance, with the communication energy consumption model adopted in [5] and the assumption that sensors have equal amounts of initial energy, we will have

$$\jmath \propto \psi^{2+\alpha_{rf}}, \tag{3}$$

in which α_{rf} is the RF attenuation exponent and typically in the range of 2 to 5.

Note that the arguments of the functions in (1) and (2) are sensor locations in A, not sensor identities. Thus, (1) and (2) describe network operations using the geographic model (e.g., where the information packet should be forwarded to?) instead of the commonly used graph model (e.g., which sensor the information packet should be forwarded to?), and have no scalability problem as the number of sensors increases.

Ko [4] presents a routing algorithm, DGSI-FEM, which numerically solves $\mathbf{D}$ from (2) for the routing direction. DGSI-FEM first discretizes continuous functions involved in (2) by finite element method (FEM) [6] and leads to a set of linear equations, and then uses the distributed Gauss-Seidel iteration (DGSI) algorithm to solve the set of linear equations by the Gauss-Seidel iteration (GSI) method.

However DGSI coordinates sensors to update the values of unknowns in lexicographical order, and some sensors may fail to collect the information regarding solution precision due to the presence of holes (i.e., sensor-less areas in ROI), and thus terminate with fewer iterations, resulting in larger errors. Thus, we propose in this paper ARB-DGSI-FEM algorithm, which not only eliminates the early termination problem, but also strives for the greatest degree of parallelism to reduce the convergence time.

II. Preliminaries

A. Finite Element Method (FEM)

Equation (2) can be solved numerically by FEM in which (2) is locally approximated (posed over small partitions called elements of the entire ROI) and a global solution is built by combining the local solutions over these elements [6]. Referring to Fig. 1, we may divide the ROI into equally-spaced grids and then use these grid points to form the elements (e.g., the gray hexagon on the $x-y$ plane illustrated in Fig. 2).

Consider the set of basis functions, $\mu_{i,j}$ with $\wr i,j\int \in \mathcal{G}$, defined on the A such that $\mu_{i,j}$ has the following properties:

$$\mu_{i,j}(x_{i'},y_{j'}) = \begin{cases} 1 & \text{if } i'=i \text{ and } j'=j, \\ 0 & \text{otherwise,} \end{cases}$$

and

$$\forall (x,y)\in A,\ \sum_{\wr i,j\int\in\mathcal{G}} \mu_{i,j}(x,y) = 1. \tag{4}$$

Here $\mathcal{G}$ is the set of grid points in A and $(x_{i'},y_{j'})$ is the position of $\wr i',j'\int$. We then approximate Φ, $\mathcal{J}$, and ρ respectively by

$$\Phi(x,y) = \sum_{\wr i,j\int\in\mathcal{G}} \widetilde{\Phi}_{i,j}\mu_{i,j}(x,y), \tag{5}$$

$$\mathcal{J}(x,y) = \sum_{\wr i,j\int\in\mathcal{G}} \widetilde{\mathcal{J}}_{i,j}\mu_{i,j}(x,y), \tag{6}$$

$$\rho(x,y) = \sum_{\wr i,j\int\in\mathcal{G}} \widetilde{\rho}_{i,j}\mu_{i,j}(x,y), \tag{7}$$

in which $\widetilde{\Phi}_{i,j} = \Phi(x_i,y_j)$, $\widetilde{\mathcal{J}}_{i,j} = \mathcal{J}(x_i,y_j)$, and $\widetilde{\rho}_{i,j} = \rho(x_i,y_j)$. By substituting (5), (6), and (7) into (2), we obtain the following set of linear equations

$$\forall \wr i,j\int \in \mathcal{G},\ \sum_{\wr i',j'\int\in\mathcal{G}} K_{i,j}^{i',j'}\widetilde{\Phi}_{i',j'} = g_{i,j}. \tag{8}$$

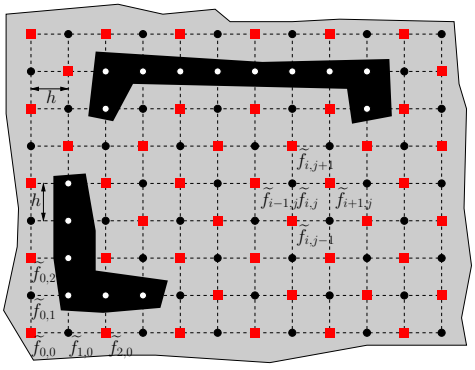


Fig. 1. The definition domain of f (e.g., Φ or $\mathbf{D}$) is divided into equally spaced grids with a grid size h. $\wr i,j\int$ is the grid point located at the i-th column and the j-th row. $\widetilde{f}_{i,j}$ is the value of f at $\wr i,j\int$. The grid points marked by white circles are not in the ROI, and the grid points marked by squares and black circles are respectively in $\mathcal{R}_\Phi^{[0]}$ and $\mathcal{R}_\Phi^{[1]}$ which are explained in Sec. III.

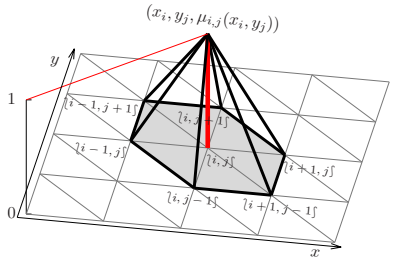


Fig. 2. The linear basis function $\mu_{i,j}$ is a pyramid with the peak at $\wr i,j\int$ and is non-zero only within the element centered at $\wr i,j\int$ (i.e., the gray hexagon). In addition, (x_i,y_j) is the position of $\wr i,j\int$.

In this paper, we use the piecewise linear functions over hexagon elements, referring to Fig. 2, as the basis functions. It is not difficult to verify such a set of piecewise linear functions satisfies (4). For the sake of brevity, the formulae of $K_{i,j}^{i',j'}$ and $g_{i,j}$ are listed in Appendix A.

After $\widetilde{\Phi}_{i,j}$s are solved from (8), $\widetilde{\mathbf{D}}_{i,j}$ may be approximated by the following formulae, derived by using (5) to approximate (1):

$$\widetilde{\mathbf{D}}_{i,j_x} = \frac{\delta_{i+1,j}\delta_{i-1,j}\widetilde{\mathcal{J}}_{i,j}}{2h}\left(\widetilde{\Phi}_{i+1,j}-\widetilde{\Phi}_{i-1,j}\right),$$

$$\widetilde{\mathbf{D}}_{i,j_y} = \frac{\delta_{i,j+1}\delta_{i,j-1}\widetilde{\mathcal{J}}_{i,j}}{2h}\left(\widetilde{\Phi}_{i,j+1}-\widetilde{\Phi}_{i,j-1}\right),$$

in which $\widetilde{\mathbf{D}}_{i,j_x}$ and $\widetilde{\mathbf{D}}_{i,j_y}$ are respectively the x and y components of $\widetilde{\mathbf{D}}_{i,j}$. In addition, $\delta_{i,j}=1$ if $\wr i,j\int\in\mathcal{G}$, or 0 otherwise. Once $\widetilde{\mathbf{D}}_{i,j}$ is computed, its direction can be used as guidance to find the next forwarding node.

B. Distributed Gauss-Seidel Iteration (DGSI)

DGSI [7] coordinates sensors to solve (8) via GSI that iteratively updates each $\widetilde{\Phi}$ in parallel from the recent values of other $\widetilde{\Phi}$s in an order specified by $\mathcal{O}_L(i,j)$ (i.e., $\widetilde{\Phi}_{i,j}$ with smaller $\mathcal{O}_L(i,j)$ will be updated earlier). That is, $\widetilde{\Phi}_{i,j}^{(k)}$ are computed sequentially by

$$\widetilde{\Phi}_{i,j}^{(k)} \leftarrow \frac{1}{K_{i,j}^{i,j}}\left(g_{i,j} - \sum_{\substack{\mathcal{O}_L(i',j')\\ <\mathcal{O}_L(i,j)}} K_{i,j}^{i',j'}\widetilde{\Phi}_{i',j'}^{(k)} - \sum_{\substack{\mathcal{O}_L(i',j')\\ >\mathcal{O}_L(i,j)}} K_{i,j}^{i',j'}\widetilde{\Phi}_{i',j'}^{(k-1)}\right), \tag{9}$$

in which $\mathcal{O}_L(i,j)$ defines the lexicographical order; that is, $\mathcal{O}_L(i_1,j_1) < \mathcal{O}_L(i_2,j_2)$ if $i_1<i_2$, or $i_1=i_2$ and $j_1<j_2$.

We say GSI has gone through one *sweep* when all $\widetilde{\Phi}$s update once, and $\widetilde{\Phi}_{i,j}^{(k)}$ is the $\widetilde{\Phi}$ value of $\wr i,j\int$ after the k-th sweep. Referring to Appendix A, $K_{i,j}^{i',j'}=0$ if $\wr i',j'\int \neq \wr i,j\int$ and $\wr i',j'\int \notin \mathcal{N}_{i,j}^1$, in which $\mathcal{N}_{i,j}^1$ is the set of adjacent grid points of $\wr i,j\int$ in A. Thus, only $\widetilde{\Phi}_{i,j-1}^{(k)}$, $\widetilde{\Phi}_{i-1,j}^{(k)}$, $\widetilde{\Phi}_{i+1,j}^{(k-1)}$, and $\widetilde{\Phi}_{i,j+1}^{(k-1)}$ are needed to compute $\widetilde{\Phi}_{i,j}^{(k)}$ via (9). In other words, as long as $\widetilde{\Phi}_{i,j-1}^{(k)}$ and $\widetilde{\Phi}_{i-1,j}^{(k)}$ are computed, $\widetilde{\Phi}_{i,j}^{(k)}$ can be computed.

The iteration (9) will continue until $\left|\widetilde{\Phi}_{i,j}^{(k)} - \widetilde{\Phi}_{i,j}^{(k-1)}\right| \leq \varepsilon$ for all $\wr i,j\int \in \mathcal{G}$. DGSI adopts a two-phase approach (namely, a forward sweep followed by a backward sweep) to collect and broadcast convergence and termination information. We denote $\mathcal{N}_{i,j}^{1,\mathcal{O}_L<}$ and $\mathcal{N}_{i,j}^{1,\mathcal{O}_L>}$ respectively as the sets of the "smaller" and "greater" (in the order specified by $\mathcal{O}_L$) adjacent points of $\wr i,j\int$ in A, i.e., $\mathcal{N}_{i,j}^{1,\mathcal{O}_L<} = \left\{\wr i',j'\int \in \mathcal{N}_{i,j}^{1} \middle| \mathcal{O}_L(i',j') < \mathcal{O}_L(i,j)\right\}$ and $\mathcal{N}_{i,j}^{1,\mathcal{O}_L>} = \left\{\wr i',j'\int \in \mathcal{N}_{i,j}^{1} \middle| \mathcal{O}_L(i',j') > \mathcal{O}_L(i,j)\right\}$. In addition, we say $\wr i,j\int$ is an *$\mathcal{O}_L$-initiator* if $\mathcal{N}_{i,j}^{1,\mathcal{O}_L<} = \varnothing$, or an *$\mathcal{O}_L$-terminator* if $\mathcal{N}_{i,j}^{1,\mathcal{O}_L>} = \varnothing$. Hence, DGSI for $\wr i,j\int$[1] will proceed in the following steps after initialization, referring to Fig. 3 for the sequence diagram of DGSI:

1) Forward sweep: iteration begins from the $\mathcal{O}_L$-initiator (e.g., the bottom-left grid point) to the $\mathcal{O}_L$-terminator (e.g., the top-right grid point).
   a) waits for $\widetilde{\Phi}$s and DONEs from all non-terminated grid points in $\mathcal{N}_{i,j}^{1,\mathcal{O}_L<}$;
   b) updates $\widetilde{\Phi}_{i,j}$ by (9) and $\text{DONE}_{i,j}$;
   c) sends $\widetilde{\Phi}_{i,j}$ and $\text{DONE}_{i,j}$ to all grid points in $\mathcal{N}_{i,j}^{1,\mathcal{O}_L>}$;
   d) terminates if $\text{DONE}_{i,j}$ is **true**.
2) Backward sweep: iteration begins from the $\mathcal{O}_L$-terminator back to the $\mathcal{O}_L$-initiator.
   a) waits for $\widetilde{\Phi}$s and PRECISEs from all grid points in $\mathcal{N}_{i,j}^{1,\mathcal{O}_L>}$;
   b) updates $\widetilde{\Phi}_{i,j}$ by (9) and $\text{PRECISE}_{i,j}$;
   c) sends $\widetilde{\Phi}_{i,j}$ and $\text{PRECISE}_{i,j}$ to all grid points in $\mathcal{N}_{i,j}^{1,\mathcal{O}_L<}$.

The convergence and termination information are propagated via $\text{PRECISE}_{i,j}$ and $\text{DONE}_{i,j}$ which indicate whether the update changes are small enough and whether the iteration should terminate respectively. $\wr i,j\int$ will set $\text{PRECISE}_{i,j}$ as **true** if both the following two conditions are satisfied:

1) $\left|\widetilde{\Phi}_{i,j}^{(k)} - \widetilde{\Phi}_{i,j}^{(k-1)}\right| \leq \varepsilon$;
2) $\forall \wr i',j'\int \in \mathcal{N}_{i,j}^{1,\mathcal{O}_L>}, \text{PRECISE}_{i',j'} = \textbf{true}$.

Thus, the convergence information is aggregated, via PRECISEs, starting from the $\mathcal{O}_L$-terminator in the backward sweep. The PRECISEs collected by the $\mathcal{O}_L$-initiator at the end of the backward sweep indicate whether the update changes of all $\widetilde{\Phi}$s are small enough, and are used to determine the termination by the $\mathcal{O}_L$-initiator. That is, $\wr i,j\int$ will set $\text{DONE}_{i,j}$ as **true** based on the following rules:

1) $\wr i,j\int$ is an $\mathcal{O}_L$-initiator and $\text{PRECISE}_{i,j} = \textbf{true}$;
2) $\wr i,j\int$ is not an $\mathcal{O}_L$-initiator but $\forall \wr i',j'\int \in \mathcal{N}_{i,j}^{1,\mathcal{O}_L<}, \text{DONE}_{i',j'} = \textbf{true}$.

[1]In DGSI, a nearby sensor is selected as the *grid head* for each grid point to compute the value of Φ. For the sake of brevity, we simply describe the operations of grid points without explicitly mentioning that the operations are actually executed by grid heads.

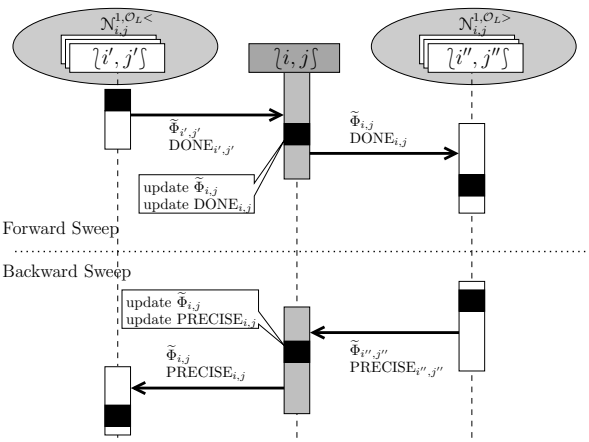


Fig. 3. Sequence diagram for $\wr i,j\int$ in the iteration of DGSI: for example, in the forward sweep, the grid points in $\mathcal{N}_{i,j}^{1,\mathcal{O}_L<}$ (i.e., $\wr i,j-1\int$ and $\wr i-1,j\int$) update and then send their $\widetilde{\Phi}$s and DONEs to $\wr i,j\int$. After updating $\widetilde{\Phi}_{i,j}$ and $\text{DONE}_{i,j}$, $\wr i,j\int$ sends $\widetilde{\Phi}_{i,j}$ and $\text{DONE}_{i,j}$ to the grid points in $\mathcal{N}_{i,j}^{1,\mathcal{O}_L>}$ (i.e., $\wr i+1,j\int$ and $\wr i,j+1\int$).

III. Atomic Red-Black Distributed Gauss-Seidel Routing with Finite Element Method (ARB-DGSR-FEM) Algorithm

We have Theorem 1 which states that if $\wr i,j\int$ has the knowledge of whether $\left|\widetilde{\Phi}_{i',j'}^{(k)} - \widetilde{\Phi}_{i',j'}^{(k-1)}\right| \leq \varepsilon$, there must exist a *backward path* from $\wr i',j'\int$ to $\wr i,j\int$. Note that though DGSI described in the previous section uses $\mathcal{O}_L$, DGSI may work with different orders and thus the following definition and theorem do not focus on any specific order. For the sake of brevity, the proof of Theorem 1 is given in Appendix B.

Definition 1. A *backward path* from $\wr i_s,j_s\int$ to $\wr i_d,j_d\int$ is an ordered sequence of grid points, $\mathcal{P} = \langle \wr i_0,j_0\int, \wr i_1,j_1\int, \ldots, \wr i_p,j_p\int\rangle$ with $\wr i_0,j_0\int = \wr i_s,j_s\int$ and $\wr i_p,j_p\int = \wr i_d,j_d\int$, such that $\wr i_{k+1},j_{k+1}\int \in \mathcal{N}_{i_k,j_k}^{1,\mathcal{O}<}$. Here the order of a grid point is specified by $\mathcal{O}$.

Theorem 1. *By the rule that DGSI sets the value of $\text{PRECISE}_{i,j}$, $\text{PRECISE}_{i,j} =$ **true** at the k-th sweep if and only if $\left|\widetilde{\Phi}_{i,j}^{(k)} - \widetilde{\Phi}_{i,j}^{(k-1)}\right| \leq \varepsilon$ and $\left|\widetilde{\Phi}_{i',j'}^{(k)} - \widetilde{\Phi}_{i',j'}^{(k-1)}\right| \leq \varepsilon$ for all $\wr i',j'\int$s which have a backward path to $\wr i,j\int$.*

Note that an $\mathcal{O}_L$-initiator is responsible for determining the value of its DONE based on the PRECISEs it collected. However, the $\mathcal{O}_L$-initiator may fail to collect the convergence information of the grid points which have no backward path to it due to the presence of holes. Referring to Fig. 4, $\wr 0,0\int$ may terminate the iteration by setting $\text{DONE}_{0,0} =$ **true** without knowing whether $\widetilde{\Phi}$s of the grid points enclosed in the darker rectangle region converge, resulting in larger errors. Similarly, $\wr 2,2\int$ may terminate the iteration without knowing whether $\widetilde{\Phi}$s of the grid points enclosed in the lighter region converge.

To eliminate this early termination problem, we may define an iteration order, $\mathcal{O}_G(i,j)$, based on the shortest path between

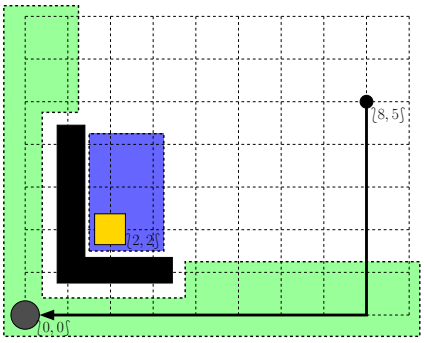


Fig. 4. $\lceil 0,0 \rfloor$ is an $\mathcal{O}_L$-initiator since $\mathcal{N}_{0,0}^{1,\mathcal{O}_L<} = \varnothing$. The directional path from $\lceil 8,5 \rfloor$ to $\lceil 0,0 \rfloor$ is a backward path. Due to the presence of the L-shaped hole, there exists the second $\mathcal{O}_L$-initiator, $\lceil 2,2 \rfloor$. The grid points enclosed in the darker rectangle region have no backward path to $\lceil 0,0 \rfloor$, and the grid points enclosed in the lighter region have no backward path to $\lceil 2,2 \rfloor$.

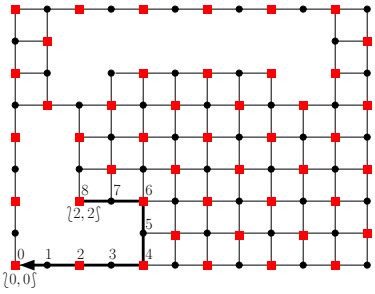


Fig. 5. $G^{\mathcal{G}}$ of Fig. 1. The thick path from $\lceil 2,2 \rfloor$ to $\lceil 0,0 \rfloor$ is the shortest path and the numbers by the grid points on the path are the $\mathcal{O}_G$ orders of these grid points.

two grid points. Referring to Fig. 5, let $G^{\mathcal{G}} = (V, E)$ be a graph such that V is the set of grid points in the ROI and there is an edge connecting $\lceil i,j \rfloor$ and $\lceil i',j' \rfloor$ if $\lceil i,j \rfloor$ and $\lceil i',j' \rfloor$ are adjacent to each other. Then $\mathcal{O}_G(i,j)$ is defined as the length of the shortest path from $\lceil i,j \rfloor$ to $\lceil 0,0 \rfloor$ under $G^{\mathcal{G}}$. It is easy to prove that the shortest path from $\lceil i,j \rfloor$ to $\lceil 0,0 \rfloor$ is also a backward path, which will guarantee that $\lceil 0,0 \rfloor$, the $\mathcal{O}_G$-initiator, can collect the convergence information of all grid points. For example, the thick path illustrated in Fig. 5 is a backward path from $\lceil 2,2 \rfloor$ to $\lceil 0,0 \rfloor$ with $\mathcal{O}_G$, but $\lceil 2,2 \rfloor$ has no backward path to $\lceil 0,0 \rfloor$ with $\mathcal{O}_L$, as depicted in Fig. 4.

We say DGSI has gone through one *round* when all grid points which can update their $\widetilde{\Phi}$s simultaneously finish updating their $\widetilde{\Phi}$s. Then the following theorem states that a complete sweep of DGSI with $\mathcal{O}_G$ may require up to $(\max \mathcal{O}_G + 1)$ rounds, in which $\max \mathcal{O}_G$ is the length of the longest shortest path to $\lceil 0,0 \rfloor$ in $G^{\mathcal{G}}$. For the sake of brevity, the proof of Theorem 2 is given in Appendix C.

Theorem 2. *In DGSI with $\mathcal{O}_G$, if we say $\lceil 0,0 \rfloor$ updates $\widetilde{\Phi}_{0,0}$ at the first round of the forward sweep, $\lceil i,j \rfloor$ updates $\widetilde{\Phi}_{i,j}$ at the $(\mathcal{O}_G(i,j)+1)$-th round.*

The parallelism of computing (9) may be enhanced by using the *red-black* order [8], denoted as $\mathcal{O}_{RB}(i,j) \stackrel{\text{def}}{=} (i+j) \bmod 2$. By denoting $\mathcal{R}_\Phi^{[0]}$ and $\mathcal{R}_\Phi^{[1]}$ as the sets of the red grid points and the black grid points respectively as illustrated in Fig. 1, i.e.,

$$\mathcal{R}_\Phi^{[n]} \stackrel{\text{def}}{=} \{\lceil i,j \rfloor \in \mathcal{G} | \mathcal{O}_{RB}(i,j) = n\},$$

it is easy to prove that if two different grid points, $\lceil i,j \rfloor$ and $\lceil i',j' \rfloor$, have the same color, $K_{i,j}^{i',j'} = 0$ and $K_{i',j'}^{i,j} = 0$ since $\lceil i',j' \rfloor \notin \mathcal{N}_{i,j}^1$. Thus all grid points with the same color can update their Φs simultaneously, and a sweep needs only two rounds.

However for DGSI with $\mathcal{O}_{RB}$, all the red grid points are $\mathcal{O}_{RB}$-initiators and any two red grid points have no backward path to each other. Thus, each black grid point has no backward path to any red grid points except its adjacent ones; in other words, each red grid point can only collect convergence information from its adjacent black grid points.

$\mathcal{O}_G$ works better for convergence but not parallelism, and the other way around for $\mathcal{O}_{RB}$. Thus we propose ARB-DGSI-FEM in which $\widetilde{\Phi}$s are updated using $\mathcal{O}_{RB}$ and the convergence information are propagated using $\mathcal{O}_G$. In addition, we simplify the two-phase coordination mechanism by introducing a new state packet, $\mathbb{S}$, to replace PRECISE and DONE used in DGSI. $\mathbb{S}_{i,j}$ represents the iteration state of $\lceil i,j \rfloor$ and may be one of three states, namely `INTERIM`, `PRECISE`, and `DONE`. Fig. 6 depicts the state transition diagram of $\mathbb{S}_{i,j}$ in which the transition conditions **T1** and **T2** are

- **T1** $\left|\widetilde{\Phi}_{i,j}^{(k)} - \widetilde{\Phi}_{i,j}^{(k-1)}\right| \le \varepsilon$ and $\forall \lceil i',j' \rfloor \in \mathcal{N}_{i,j}^{1,\mathcal{O}_G>}, \mathbb{S}_{i',j'} =$ `PRECISE`;
- **T2** $\lceil i,j \rfloor$ is an $\mathcal{O}_G$-initiator or $\forall \lceil i',j' \rfloor \in \mathcal{N}_{i,j}^{1,\mathcal{O}_G<}, \mathbb{S}_{i',j'} =$ `DONE`.

Fig. 6. Transition diagram of $\mathbb{S}$

Referring to Fig. 7 for the sequence diagram, ARB-DGSI-FEM for $\lceil i,j \rfloor$ will proceed in the following steps after initialization. Note that the iteration begins from the $\mathcal{O}_{RB}$-initiator (i.e., red grid points) to the $\mathcal{O}_{RB}$-terminator (i.e., black grid points).

1) waits for $\widetilde{\Phi}$s and $\mathbb{S}$s from all non-terminated adjacent grid points;
2) updates $\widetilde{\Phi}_{i,j}$ by (9) with $\mathcal{O}_L$ replaced by $\mathcal{O}_{RB}$;
3) updates $\mathbb{S}_{i,j}$ based on **T1** and **T2**;
4) sends $\widetilde{\Phi}_{i,j}$ and $\mathbb{S}_{i,j}$ to all adjacent grid points;
5) terminates if $\mathbb{S}_{i,j} =$ `DONE`.

T1 implies that $\lceil i,j \rfloor$ determines whether $\mathbb{S}_{i,j}$ will transit to `PRECISE` based on the state information from the grid points in $\mathcal{N}_{i,j}^{1,\mathcal{O}_G>}$. Similarly, from **T2**, $\lceil i,j \rfloor$ determines whether $\mathbb{S}_{i,j}$ will transit to `DONE` based on the state information from the grid points in $\mathcal{N}_{i,j}^{1,\mathcal{O}_G<}$. Thus, though both $\widetilde{\Phi}$s and $\mathbb{S}$s are transmitted together between adjacent grid points, $\widetilde{\Phi}$s are updated in $\mathcal{O}_{RB}$ but the iteration is coordinated similar to DGSI with $\mathcal{O}_G$; that is, convergence information ($\mathbb{S} =$

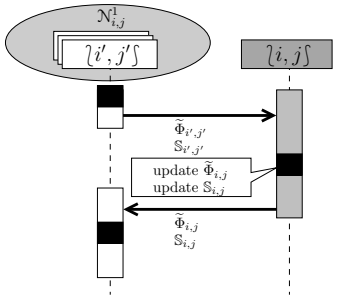
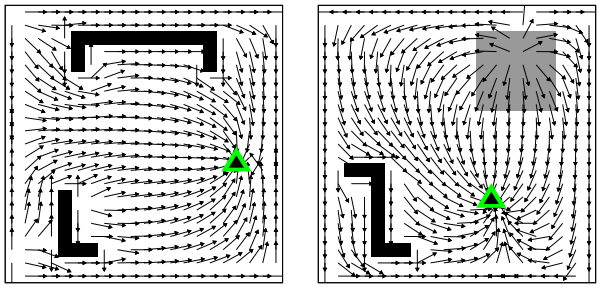


Fig. 7. Sequence diagram for $\wr i, j \int$ in the iteration of ARB-DGSI-FEM: all the grid points in $\mathcal{N}^1_{i,j}$ update and then send their $\widetilde{\Phi}$s and $\mathbb{S}$s to $\wr i, j \int$. After updating $\widetilde{\Phi}_{i,j}$ and $\mathbb{S}_{i,j}$, $\wr i, j \int$ sends $\widetilde{\Phi}_{i,j}$ and $\mathbb{S}_{i,j}$ to the grid points in $\mathcal{N}^1_{i,j}$.



(a) Uniform ψ and ρ

(b) Uniform ψ and nonuniform ρ: the sensors in the gray region generate ten times more information than other sensors.

Fig. 8. Routing directions obtained by ARB-DGSI-FEM: the sink is marked by a triangle and the hole is represented as a black region.

PRECISE) is aggregated starting from the $\mathcal{O}_G$-terminator to the $\mathcal{O}_G$-initiator, and termination information ($\mathbb{S} = $ DONE) is propagated from the $\mathcal{O}_G$-initiator to the $\mathcal{O}_G$-terminator.

IV. Numerical results

We present two numerical examples to demonstrate the effectiveness of ARB-DGSI-FEM. Each ROI is divided into 19×19 grids. The sensors are randomly deployed with the constant density distribution ψ, and generate information based on ρ except for the sink which will consume all the information generated. In addition, $\mathcal{I}$ is a constant due to the constant density distribution by (3).

The routing directions obtained by ARB-DGSI-FEM are depicted as arrows in Fig. 8. The directions indicate that information may be forwarded in a direction which deviates from a straight line to the sink to bypass the holes in advance to avoid the excess energy consumption of the boundary sensors. In addition, as illustrated in Fig. 8(b), the sensors along the top side of the gray region may forward information in the opposite direction to the sink in order to avoid using sensors in the high-ρ region for load balancing.

Table I presents the comparative performance results for DGSI, DMSR [9], and ARB-DGSI-FEM. Here DMSR allows an arbitrary grid point other than $\wr 0, 0 \int$ to be assigned as the $\mathcal{O}_L$-initiator to minimize the length of the longest backward path for better parallelism. All three algorithms require similar number of sweeps.[2] However, ARB-DGSI-FEM can significantly improve the performance since ARB-DGSI-FEM has a higher degree of parallelism; ARB-DGSI-FEM requires less than 2% of rounds and thus the convergence time needed by DGSI, or less than 12% of rounds and time needed by DMSR.

TABLE I
Parallelism and errors for the ROI illustrated in Fig. 8

Fig. 8(a)	DGSI	DMSR	ARB-DGSI-FEM
sweeps	1893/7885	1933	1963
rounds per sweep	38	20	2
total rounds	299,630	38,660	3,926
relative residual (%)	1.37×10^{-6}	1.48×10^{-6}	1.55×10^{-6}
Fig. 8(b)			
sweeps	1887/6943	2227	2301
rounds per sweep	38	18	2
total rounds	263,834	40,086	4,602
relative residual (%)	5.29×10^{-6}	6.54×10^{-6}	5.86×10^{-6}

Table I also compares the accuracy of these algorithms. Here the error is measured by the *relative residual*, defined as $\|\mathbf{K}\widetilde{\mathbf{\Phi}} - \mathbf{g}\|/\|\mathbf{g}\|$, in which $\mathbf{K}$, $\widetilde{\mathbf{\Phi}}$, and $\mathbf{g}$ are respectively the matrix forms of $K^{i',j'}_{i,j}$, $\widetilde{\Phi}_{i,j}$, and $g_{i,j}$ in (8). The relative residual indicates how close to the exact solution of (8) the current $\widetilde{\Phi}$s are. The results show that the simpler one-phase coordination mechanism of ARB-DGSI-FEM does not sacrifice much accuracy.

V. Conclusion

DGSI coordinates sensors to update $\widetilde{\Phi}$s and propagate convergence information in $\mathcal{O}_L$, which may lead to the early termination problem due to the presence of holes. We propose ARB-DGSI-FEM algorithm in which the convergence information is propagated in $\mathcal{O}_G$ to eliminate the early termination problem and $\widetilde{\Phi}$s are updated using $\mathcal{O}_{RB}$ for the maximum degree of parallelism. Our numerical results reveal that, even with a simpler one-phase coordination mechanism, ARB-DGSI-FEM obtains more than 98% savings in the total number of rounds without too much sacrifice of accuracy.

Appendix A
Formulae of $K^{i',j'}_{i,j}$ and $g_{i,j}$

$$\begin{aligned} K^{i,j}_{i,j} = 1/6 \Big(& \mathcal{B}^0[\mathcal{I}]_{i,j\Delta^{i+1,j}_{i+1,j-1}} + \mathcal{B}^0[\mathcal{I}]_{i,j\Delta^{i+1,j-1}_{i,j-1}} \\ & + 2\mathcal{B}^0[\mathcal{I}]_{i,j\Delta^{i,j-1}_{i-1,j}} + \mathcal{B}^0[\mathcal{I}]_{i,j\Delta^{i-1,j}_{i-1,j+1}} \\ & + \mathcal{B}^0[\mathcal{I}]_{i,j\Delta^{i-1,j+1}_{i,j+1}} + 2\mathcal{B}^0[\mathcal{I}]_{i,j\Delta^{i,j+1}_{i+1,j}} \Big), \\ K^{i+1,j}_{i,j} = -1/6 \Big(& \mathcal{B}^0[\mathcal{I}]_{i,j\Delta^{i,j+1}_{i+1,j}} + \mathcal{B}^0[\mathcal{I}]_{i,j\Delta^{i+1,j}_{i+1,j-1}} \Big), \end{aligned}$$

[2]Note that some grid points may terminate earlier (i.e., less number of sweeps) in DGSI due to the existence of the hole as described in Sec. II-B.

$$K_{i,j}^{i,j-1} = -1/6\left(\mathcal{B}^0[\mathcal{I}]_{i,j\Delta_{i,j-1}^{i+1,j-1}} + \mathcal{B}^0[\mathcal{I}]_{i,j\Delta_{i-1,j}^{i,j-1}}\right),$$

$$K_{i,j}^{i-1,j} = -1/6\left(\mathcal{B}^0[\mathcal{I}]_{i,j\Delta_{i-1,j}^{i,j-1}} + \mathcal{B}^0[\mathcal{I}]_{i,j\Delta_{i-1,j+1}^{i-1,j}}\right),$$

$$K_{i,j}^{i,j+1} = -1/6\left(\mathcal{B}^0[\mathcal{I}]_{i,j\Delta_{i,j+1}^{i-1,j+1}} + \mathcal{B}^0[\mathcal{I}]_{i,j\Delta_{i+1,j}^{i,j+1}}\right),$$

$$K_{i,j}^{i',j'} = 0, \text{ if } \wr i',j'\int \neq \wr i,j\int \text{ and } \wr i',j'\int \notin \mathcal{N}_{i,j}^1,$$

and

$$g_{i,j} = -h^2/24\left(\mathcal{B}^1[\rho]_{i,j\Delta_{i+1,j-1}^{i+1,j}} + \mathcal{B}^1[\rho]_{i,j\Delta_{i,j-1}^{i+1,j-1}} + \mathcal{B}^1[\rho]_{i,j\Delta_{i-1,j}^{i,j-1}} + \mathcal{B}^1[\rho]_{i,j\Delta_{i-1,j+1}^{i-1,j}} + \mathcal{B}^1[\rho]_{i,j\Delta_{i,j+1}^{i-1,j+1}} + \mathcal{B}^1[\rho]_{i,j\Delta_{i+1,j}^{i,j+1}}\right),$$

in which

$$\mathcal{B}^0[f]_{i,j\Delta_{i_2,j_2}^{i_1,j_1}} = \delta_{i_1,j_1}\delta_{i_2,j_2}\left(\widetilde{f}_{i,j} + \widetilde{f}_{i_1,j_1} + \widetilde{f}_{i_2,j_2}\right),$$

$$\mathcal{B}^1[f]_{i,j\Delta_{i_2,j_2}^{i_1,j_1}} = \delta_{i_1,j_1}\delta_{i_2,j_2}\left(2\widetilde{f}_{i,j} + \widetilde{f}_{i_1,j_1} + \widetilde{f}_{i_2,j_2}\right).$$

APPENDIX B
PROOF OF THEOREM 1

As described in Sec. II-B, $\wr i,j\int$ will set $\text{PRECISE}_{i,j}$ as **true** at the k-th sweep if the following two conditions are satisfied:

C-1 $\left|\widetilde{\Phi}_{i,j}^{(k)} - \widetilde{\Phi}_{i,j}^{(k-1)}\right| \leq \varepsilon$;

C-2 $\forall \wr i',j'\int \in \mathcal{N}_{i,j}^{1,\mathcal{O}>}, \text{PRECISE}_{i',j'} = \textbf{true}$.

Here the order of a grid point is specified by $\mathcal{O}$.

($\Rightarrow$)

Note that **C-1** guarantees that $\left|\widetilde{\Phi}_{i,j}^{(k)} - \widetilde{\Phi}_{i,j}^{(k-1)}\right| \leq \varepsilon$. Thus we only need to prove that $\left|\widetilde{\Phi}_{i',j'}^{(k)} - \widetilde{\Phi}_{i',j'}^{(k-1)}\right| \leq \varepsilon$ for all $\wr i',j'\int$s which have a backward path to $\wr i,j\int$, which can be easily done by the following lemma.

Lemma 1. *Suppose $\wr i^*,j^*\int$ has a backward path $\mathcal{P}$ to $\wr i,j\int$ and $\left|\widetilde{\Phi}_{i^*,j^*}^{(k)} - \widetilde{\Phi}_{i^*,j^*}^{(k-1)}\right| > \varepsilon$. If $\wr i',j'\int$ is on $\mathcal{P}$, $\text{PRECISE}_{i',j'} =$ **false**.*

Proof. We prove this lemma by induction on the grid points of $\mathcal{P}$. Let $\mathcal{P} = \langle \wr i_0,j_0\int, \wr i_1,j_1\int, \ldots, \wr i_p,j_p\int\rangle$ with $\wr i_0,j_0\int = \wr i^*,j^*\int$ and $\wr i_p,j_p\int = \wr i,j\int$. Obviously, $\left|\widetilde{\Phi}_{i^*,j^*}^{(k)} - \widetilde{\Phi}_{i^*,j^*}^{(k-1)}\right| > \varepsilon$ leads to $\text{PRECISE}_{i_0,j_0} =$ **false** by **C-1**.

Suppose $\text{PRECISE}_{i_n,j_n} =$ **false** for $0 \leq n < p$. By Definition 1, $\wr i_{n+1},j_{n+1}\int \in \mathcal{N}_{i_n,j_n}^{1,\mathcal{O}<}$ or $\wr i_n,j_n\int \in \mathcal{N}_{i_{n+1},j_{n+1}}^{1,\mathcal{O}>}$. Since $\text{PRECISE}_{i_n,j_n} =$ **false**, $\text{PRECISE}_{i_{n+1},j_{n+1}} =$ **false** by **C-2**. □

If $\wr i',j'\int$ has a backward path $\mathcal{P}$ to $\wr i,j\int$ and $\left|\widetilde{\Phi}_{i',j'}^{(k)} - \widetilde{\Phi}_{i',j'}^{(k-1)}\right| > \varepsilon$, the value of PRECISE of each grid point, including $\wr i,j\int$, on $\mathcal{P}$ is **false**, which contradicts to $\text{PRECISE}_{i,j} =$ **true**.

($\Leftarrow$)

Suppose $\text{PRECISE}_{i,j} =$ **false**. Since $\left|\widetilde{\Phi}_{i,j}^{(k)} - \widetilde{\Phi}_{i,j}^{(k-1)}\right| \leq \varepsilon$, $\exists \wr i_1,j_1\int \in \mathcal{N}_{i,j}^{1,\mathcal{O}>}, \text{PRECISE}_{i_1,j_1} =$ **false** by **C-2**. Obviously, $\langle \wr i_1,j_1\int, \wr i,j\int\rangle$ is a backward path from $\wr i_1,j_1\int$ to $\wr i,j\int$. Thus, $\left|\widetilde{\Phi}_{i_1,j_1}^{(k)} - \widetilde{\Phi}_{i_1,j_1}^{(k-1)}\right| \leq \varepsilon$, and $\exists \wr i_2,j_2\int \in \mathcal{N}_{i_1,j_1}^{1,\mathcal{O}>}, \text{PRECISE}_{i_2,j_2} =$ **false** by **C-2**. Similarly, $\langle \wr i_2,j_2\int, \wr i_1,j_1\int, \wr i,j\int\rangle$ is a backward path from $\wr i_2,j_2\int$ to $\wr i,j\int$. Thus, $\left|\widetilde{\Phi}_{i_2,j_2}^{(k)} - \widetilde{\Phi}_{i_2,j_2}^{(k-1)}\right| \leq \varepsilon$, and $\exists \wr i_3,j_3\int \in \mathcal{N}_{i_2,j_2}^{1,\mathcal{O}>}, \text{PRECISE}_{i_3,j_3} =$ **false** by **C-2**. In addition, $\langle \wr i_3,j_3\int, \wr i_2,j_2\int, \wr i_1,j_1\int, \wr i,j\int\rangle$ is a backward path from $\wr i_3,j_3\int$ to $\wr i,j\int$.

We can continue this process until we have a backward path $\langle \wr i_n,j_n\int, \wr i_{n-1},j_{n-1}\int, \ldots, \wr i,j\int\rangle$ such that $\text{PRECISE}_{i_n,j_n} =$ **false** and $\mathcal{N}_{i_n,j_n}^{1,\mathcal{O}>} = \varnothing$. Since $\mathcal{N}_{i_n,j_n}^{1,\mathcal{O}>} = \varnothing$, **C-2** is not applicable and thus $\left|\widetilde{\Phi}_{i_n,j_n}^{(k)} - \widetilde{\Phi}_{i_n,j_n}^{(k-1)}\right| > \varepsilon$ by **C-1**, which contradicts to that $\left|\widetilde{\Phi}_{i_n,j_n}^{(k)} - \widetilde{\Phi}_{i_n,j_n}^{(k-1)}\right| \leq \varepsilon$ since there is a backward path from $\wr i_n,j_n\int$ to $\wr i,j\int$.

APPENDIX C
PROOF OF THEOREM 2

Note that the shortest path from $\wr i,j\int$ to $\wr 0,0\int$ is also a backward path, which is denoted as $\mathcal{P} = \langle \wr i_p,j_p\int, \wr i_{p-1},j_{p-1}\int, \ldots, \wr i_0,j_0\int\rangle$ with $\wr i_p,j_p\int = \wr i,j\int$ and $\wr i_0,j_0\int = \wr 0,0\int$. In addition, $\mathcal{O}_G(i_n,j_n) = n$ by definition of $\mathcal{O}_G$. We then prove Theorem 2 by induction on the grid points of $\mathcal{P}$.

It is obvious that $\wr 0,0\int$ updates $\widetilde{\Phi}_{0,0}$ at the first round of the forward sweep. Suppose $\wr i_n,j_n\int$ updates $\widetilde{\Phi}_{i_n,j_n}$ at the $(n+1)$-th round. Since DGSI coordinates grid points to update $\widetilde{\Phi}$s from smaller $\mathcal{O}_G$ to larger $\mathcal{O}_G$ in the forward sweep, $\wr i_{n+1},j_{n+1}\int$ cannot update $\widetilde{\Phi}_{i_{n+1},j_{n+1}}$ earlier than the $(n+2)$-th round. In addition, $\nexists \wr i',j'\int \in \mathcal{N}_{i_{n+1},j_{n+1}}^1$ such that $\mathcal{O}_G(i_n,j_n) = n < \mathcal{O}_G(i',j') < \mathcal{O}_G(i_{n+1},j_{n+1}) = n+1$. Thus, $\wr i_{n+1},j_{n+1}\int$ updates $\Phi_{i_{n+1},j_{n+1}}$ at the $(n+2)$-th round.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. E. Cayirci, "A Survey on Sensor Networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[2] Y. Yu, R. Govindan, and D. Estrin, "Geographical and Energy Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks," Computer Science Department, UCLA, Technical Report UCLA/CSD-TR-01-0023, May 2001.

[3] M. Chen, V. C. Leung, S. Mao, and T. Kwon, "Receiver-oriented load-balancing and reliable routing in wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 9, no. 3, pp. 405–416, 2009.

[4] R.-S. Ko, "Using Finite Element Method to Solve the Load-Balancing Routing Problem for Massively-Dense Wireless Sensor Networks," in *Proceedings of 2016 International Computer Symposium*, Chiayi, Taiwan, Dec. 2016, pp. 204–209.

[5] F. Zhao and L. Guibas, *Wireless Sensor Networks: An Information Processing Approach*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2004.

[6] T. I. Zohdi, *A Finite Element Primer for Beginners: The Basics*, ser. SpringerBriefs in Applied Sciences and Technology. New York: Springer, 2014.

[7] R.-S. Ko, "A distributed routing algorithm for sensor networks derived from macroscopic models," *Computer Networks*, vol. 55, no. 1, pp. 314–329, Jan. 2011.

[8] Y. Saad, *Iterative Methods for Sparse Linear Systems*. Philadelphia: Society for Industrial and Applied Mathematics, 2003.

[9] R.-S. Ko, "A load-balancing routing algorithm for wireless sensor networks based on domain decomposition," *Ad Hoc Networks*, vol. 30, pp. 63–83, Jul. 2015.