

Effective Congestion Reduction for IC Package Substrate Routing

SHENGHUA LIU

Tsinghua University

GUOQIANG CHEN

Magma Design Automation, Inc.

TOM TONG JING and LEI HE

University of California at Los Angeles

ROBI DUTTA

Magma Design Automation, Inc.

and

XIAN-LONG HONG

Tsinghua University

Off-chip substrate routing for high-density packages is challenging due to requirements such as high density, lack of vertical detour, non-Manhattan routing, and primarily planar routing. The existing substrate routing algorithms often result in a large number of unrouted nets that have to be routed manually. This article develops an effective yet efficient diffusion-driven method D-Router to reduce congestion. Starting with an initial routing, we develop an effective diffusion-based congestion reduction. We iteratively find a congested window and spread out connections to reduce congestion inside the window by a simulated diffusion process based on the duality between congestion and concentration. The window is released after the congestion is eliminated. Compared with the state-of-the-art substrate routing method that leads to 480 nets unrouted for ten industrial designs with a total of 6415 nets, the D-Router reduces the amount of unrouted nets

The work by S. Liu was performed during his internship with Rio Design Automation, Inc., now a part of Magma Design Automation, Inc.

This article is partially supported by the NSF CAREER award CCR-0093273, and the NSF of China No. 60720106003 and 90607001.

Authors' addresses: S. Liu, Computer Science and Technology Department, Tsinghua University, Beijing 100084, China; G. Chen, Rio Design Automation, Inc., now a part of Magma Design Automation, Inc., San Jose, CA 95110; email: gchen@Magma-DA.COM; T. T. Jing (corresponding author), L. He, Electrical Engineering Department, University of California at Los Angeles, Los Angeles, CA 90095; email: tomjing@ee.ucla.edu; R. Dutta, Rio Design Automation, Inc., now a part of Magma Design Automation, Inc., San Jose, CA 95110; X.-L. Hong, Computer Science and Technology Department, Tsinghua University, Beijing 100084, China.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© 2010 ACM 1084-4309/2010/05-ART27 \$10.00
DOI 10.1145/1754405.1754412 <http://doi.acm.org/10.1145/1754405.1754412>

ACM Transactions on Design Automation of Electronic Systems, Vol. 15, No. 3, Article 27, Pub. date: May 2010.

to 104, a reduction to the 4.6 multiple. In addition, the D-Router obtains a similar reduction on unrouted nets but runs up to 94 times faster when compared with a negotiation-based substrate routing.

Categories and Subject Descriptors: B.7.2 [**Integrated Circuits**]: Design Aids—*Placement and routing*

General Terms: Algorithms, Experimentation

Additional Key Words and Phrases: IC package, substrate, routing, routability, congestion reduction

ACM Reference Format:

Liu, S., Chen, G., Jing, T. T., He, L., Dutta, R., and Hong, X.-L. 2010. Effective congestion reduction for IC package substrate routing. *ACM Trans. Des. Autom. Electron. Syst.* 15, 3, Article 27 (May 2010), 21 pages.

DOI = 10.1145/1754405.1754412 <http://doi.acm.org/10.1145/1754405.1754412>

1. INTRODUCTION

High-density integration within a package is required by technology scaling. However, such high density makes off-chip routing a challenging task, particularly in terms of congestion reduction. The package typically uses *Ball Grid Array* (BGA) substrate and *wire bonding* or *flip-chip* [Pascariu et al. 2003] to connect a microchip to the substrate. For wire bonding dies, the I/O pads of a microchip are connected to the *bond pads* around the cavity through bonding wires. For flip-chip dies, the *Re-Distribution Layer* (RDL) routing [Fang et al. 2007, 2005] first connects the I/O pads to *bump pads*, and *escape routing* [Ozidal and Wong 2002; Wang et al. 2006; Winkler 1996; Horiuchi et al. 2000; Shi and Cheng 2006; Ozidal et al. 1995] then breaks out bump pads to the boundary of the die at the *escape break points* in the build-up or signal layers. Finally, *substrate routing* [Tsai et al. 1998; Kubo and Takahashi 2005] connects escape break points of flip-chip dies or bond pads of wire bonding dies to *solder balls* (usually in the bottom layer) of a BGA package substrate.¹ A package substrate often contains several *build-up layers* for signal routing and other build-up layers and *core layers* for Power/Ground (P/G) planes, which are thicker than build-up layers. “Substrate routing” refers to signal routing in this article.

Substrate routing can be divided into two steps: *topological routing* and *detailed routing* [Chen and Lee 1996; Dai et al. 1991a]. This article studies topological routing. Because vias can be harmful to signal integrity in high-speed signaling and vias have larger widths than wires, vertical detour is not allowed and substrate routing is primarily planar [Xiong et al. 2006], though multiple routing layers are available in the package. To accumulate high-density planar routing, non-Manhattan routing must be used.

The necessity of high density due to technology scaling, in addition to planar and non-Manhattan routing, renders substrate routing a difficult task. Compared with on-chip routers, the existing substrate routing algorithms have much lower routability and often result in a large number of unrouted nets for

¹In package substrate routing, nets are usually “two-pin” ones, of which one pin is a start-point (defined in Section 2.1) and the other one is the solder ball in the bottom layer.

manual routing. For example, a very recent substrate routing algorithm with both the best routability reported in the literature and used in a state-of-the-art commercial tool, which proposed “dynamic pushing” to tackle the routing order problem and “flexible via staggering” to improve the routability, resulted in a 3.5% net unrouted for nine industrial designs [Liu et al. 2008]. However, the congestion reduction method of iteratively avoiding routing through congested area in Liu et al. [2008] limited its advantage in routability.

This article develops a diffusion-based routing (D-Router). Starting with an initial routing without consideration for congestion constraints, we iteratively find a congested window and spread out connections to reduce congestion inside the window by a simulated diffusion process based on the duality between congestion and concentration. The window is released after the congestion is eliminated. Compared with the most recent substrate routing [Liu et al. 2008], D-Router reduces the number of unrouted nets by 4.6 times with up to 94 times runtime reduction.

The concept of diffusion has been used in placement [Ren et al. 2005] for congestion reduction. The following are differences between Ren et al. [2005] and this work. In Ren et al. [2005], the movement of overlapped cells to an uncongested area in placement is based on a specified finite time-step in a global diffusion. Global diffusion guarantees convergence but may cause unnecessary cell movement, which may result in inefficiency. The localized diffusion used in this work needs a well-designed flow for convergence, but works more efficiently than its predecessor. An analytical diffusion model was used in Ren et al. [2005]. Such an analytical diffusion model and its continuous formulation, however, cannot be used in this work since we mesh the routing plane into triangles and each triangle edge is atomic for net movement. Moreover, during net movement, we must keep the original spatial order of nets to avoid intersections, which requires a new diffusion-based method. To achieve discrete diffusion, we iteratively move one net segment or one net end-point in the end-zone (defined in Section 2.1) to adjacent triangle edges.

The earlier substrate routing Surf system [Staepelaere et al. 1993] applied topological routing to generate a rubberband sketch [Dai et al. 1991b] and transformed the sketch first to a spoke sketch and then to a precise geometrical layout. Surf assumed a fixed endpoint but our formulation uses an end-zone; a geometrical layout, however, is not considered in this article. Our topological routing formulation is more flexible and therefore increases routability. In addition, Surf completed topological routing with a global routing stage followed by a local routing. Our D-Router uses iterative congestion reduction by diffusion without partitioning and therefore avoids Surf’s problem of fixing congestion only within each bin during local routing.

To better appreciate the D-Router, we also implement a negotiation-based substrate routing, which obtains a similar routability and runtime as Liu et al. [2008] and therefore is inferior to D-Router. In addition to negotiation-based routing [McMurchie and Ebeling 1995; Roy and Markov 2007], a recent on-chip router, the BoxRouter, also achieves a good routability [Cho and Pan 2006], where all nets within a congested window are ripped up and rerouted simultaneously by an Integrity Linear Programming (ILP) method. However, the

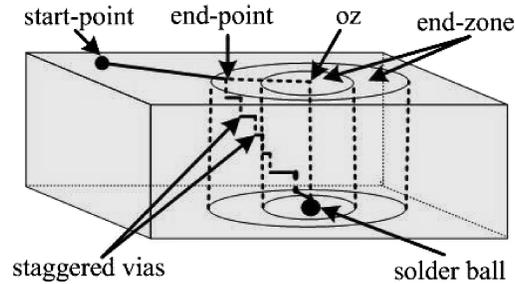


Fig. 1. End-zone and the flexible end-point.

ILP method assumes Manhattan routing, and it has not yet been shown to be compatible with non-Manhattan substrate routing. While both the BoxRouter and the D-Router reduce congestion within a window, the D-Router essentially rips up and reroutes wire segments net-by-net, whereas the BoxRouter rips up and reroutes all nets simultaneously. Additionally, the BoxRouter expands the window, but the D-Router iterates window by window.

The rest of this article is organized as follows. Section 2 formulates the routing problem. Section 3 introduces the baseline substrate topological routing methods and motivating examples. Section 4 and Section 5 introduce detailed explanations of planar diffusion and vertical diffusion in our algorithm D-Router, respectively. Section 6 presents experimental results and Section 7 concludes this article. We presented an extended abstract of this article at the 2009 International Symposium on Physical Design [Liu et al. 2009].

2. PRELIMINARIES

We use the same problem formulation and data structure as Liu et al. [2008].

2.1 Problem Formulation

We first define terms needed for problem formulation.

Start-Point. The escape break point or bond pad of a net is its *start-point*. In substrate routing, there is little freedom to do layer assignment because the escape routing [Ozidal and Wong 2002; Wang et al. 2006; Winkler 1996; Horiuchi et al. 2000; Shi and Cheng 2006; Ozidal et al. 1995] or wire bonding has already decided the layer of where the start-point of a net is located.

End-Zone. Instead of connecting a net from its start-point to a fixed end-point that is often the location above the center of the associated solder ball but in the same layer as the start-point, staggered vias allow a certain degree of flexibility in a zone defined as the end-zone. The center of the end-zone is directly on top of the destination ball. There are required offsets between staggered vias. The end-zone is a circle with radius $R_1 = \sum_i pd_i$, where pd_i is the maximal staggered via pitch in the layer with index i and a net can stop anywhere in its end-zone. The end-zone in Figure 1 also has an inner circle with radius $R_2 = \sum_i md_i$, where md_i is the minimal staggered via pitch in the layer with index i . The inner circle is defined for cost function calculations in the baseline algorithm and our initial routing presented in Liu et al. [2008].

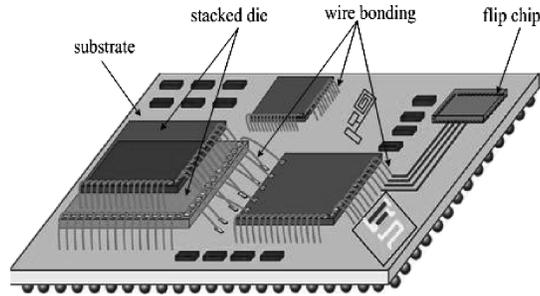


Fig. 2. An example of IC package.

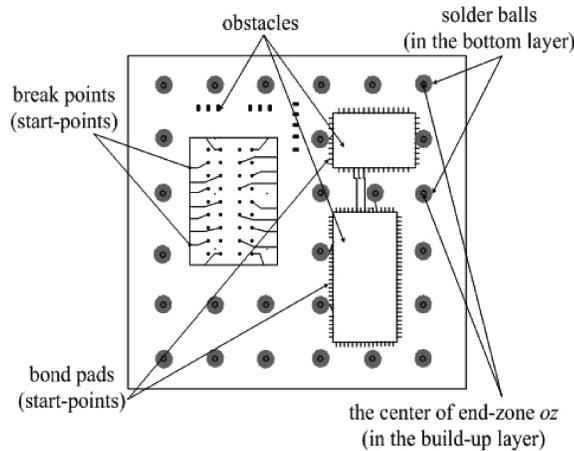


Fig. 3. Substrate Routing Graph (SRG) in a build-up layer.

The end-zone radius R_1 is also used for momentary diffusion operations to be presented in Section 4.3, case 2.

Netlist. Different from Yu and Dai [1995] and Yu et al. [1996], we assume that pin (i.e., BGA ball) assignment is given by designers. Netlist gives the definition of connections between start-points and end-zones.

Obstacles. For each build-up or signal layer, obstacles include the escape area for escape routing, wire bonding area for bonding wires, prerouted connections and vias, and other obstacles.

Figure 2 shows an example of an IC package. Assuming that the layer assignment is given by the escape routing or wire bonding, the problem of substrate routing is equivalent to a single-layer routing, which is performed on the Substrate Routing Graph (SRG) (see Figure 3) that maps the start-points, end-zones, and dies (as obstacles) on a graph. The problem of substrate topological routing for package is formulated as follows.

Formulation 1 (Substrate Topological Routing). Given start-points, end-zones, obstacles on an SRG graph, and netlist, find a topological routing solution connecting each start-point to any point in the end-zone defined by netlist, such that the routed nets satisfy the capacity constraints, have minimal wire length, and do not have vertical detour nor intersections.

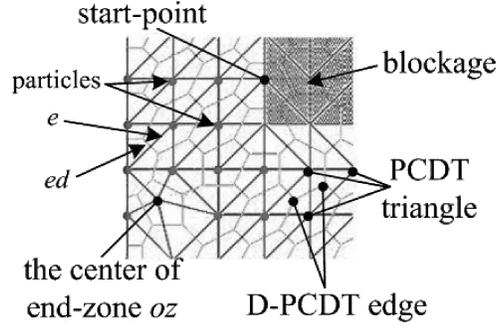


Fig. 4. Partial PCDT and D-PCDT.

2.2 Data Structure

The SRG graph for each signal layer is further discretized by a set of simple elements: triangles in two dimensions. In this article, we apply a triangle mesh by Constraint Delaunay Triangulation (CDT) [Bossen 1996], which guarantees a low computational cost and reasonably well-shaped elements. Uniformly spreading points, called *particles* U , are added in the same way as Liu et al. [2008] to the SRG plane for Particle-insertion-based CDT (PCDT) construction. Then, we build a PCDT graph based on start-points, centers of the end-zones (*ozs*), obstacles, particles U , and the boundary of the SRG plane. Thus, the start-points and the *ozs* become vertices of triangles. The Dual graph of PCDT is built accordingly, which we name D-PCDT.

A magnified view of a region of PCDT and its dual graph D-PCDT example is shown in Figure 4. In the data structure, the path of each two-pin net is along the D-PCDT. Represented in PCDT, a net path passes through common edges from one triangle to another and crossing points are assigned on triangle edges. Thus, a triangle edge constrains the number of nets passing through it. Let the capacity of edge e of PCDT be C_e . If edge e is on the boundary of an obstacle, or on the boundary of the SRG plane, then $C_e = 0$. Otherwise, $C_e = l_e$, where l_e is the length of edge e . Then, the *congestion value* η_e of edge e is defined as follows.

If C_e equals 0, edge e cannot have any net passing through it. Then, we define $\eta_e = 0$, so that these edges are not considered to reduce congestion. Otherwise,

$$\eta_e = \frac{\sum_i (w_i + s_i)}{C_e}, \quad (1)$$

where w_i and s_i are the wire segment/end-point (i.e., via) width and space of net i that passes through edge e , respectively.

3. BASELINE ALGORITHMS AND MOTIVATING EXAMPLES

3.1 Baseline Algorithms for Comparison

Liu et al. [2008] is a very recently published substrate topological routing shown in Figure 5, which is used in a commercial tool. The problem formulation is more flexible than the existing work [Tsai et al. 1998; Kubo and Takahashi

```

FOR each build-up layer of signal routing in the package,
  from top to bottom,
DO {
  s1: Initialization
  {
    Generate PCDT graph and its dual graph D-PCDT;
    Determine the end-zone for each net;
  }
  s2: Topological planar routing
  {
    WHILE ( !(congestion control achieves requirement
      && no more wire length reduction) )
    {
      S2.1: Search 2-pin net path for all nets based on
        path search algorithm
        s2.1.1 considering the end-zone;
        s2.1.2 considering congestion threshold;
      s2.2: Reorder nets;
      s2.3: Reduce congestion threshold;
    }
  }
}

```

Fig. 5. The overview of baseline algorithm.

2005]. It routes net-by-net based on A* algorithm with dynamic pushing and flexible via staggering. It also applies postrouting rip-up-and-reroute iteration for congestion reduction. It claims that good routing topology could be achieved at the beginning for routing convergence. Thus, the routing result by loosening the congestion constraint is the best choice as an initial routing for diffusion-driven congestion reduction.

We also consider negotiation-based routing, where all nets are ripped up and rerouted iteratively according to a fixed ordering and each net is rerouted with the cost function considering the current congestion and congestion history. This routing algorithm has obtained high-quality solutions to on-chip routing of FPGA [McMurchie and Ebeling 1995] and ASIC [Roy and Markov 2007; Cho et al. 2007]. For comparison in this article, we use the same substrate routing formulation from Liu et al. [2008] and enhance the A*-based algorithm from Liu et al. [2008] to reroute a net that always has two pins in substrate routing. The original cost for a node in the routing graph is

$$NC_e = rc + ec, \quad (2)$$

where rc and ec are the realized and estimated costs defined in Liu et al. [2008]. To consider congestion history required by negotiation-based routing, the new cost is defined as

$$NC'_e = (rc + h_e) \times p_e + ec, \quad (3)$$

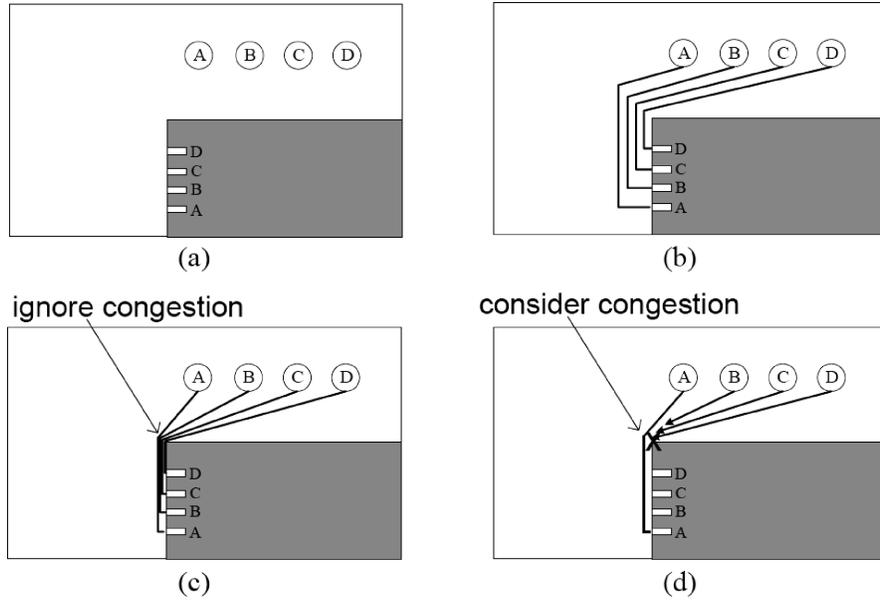


Fig. 6. An example of strong net-order dependency.

where p_e reflects the present congestion and h_e represents the congestion history. h_e is given by

$$h_e^{k+1} = \begin{cases} h_e^k + h_{inc}, & \text{if } e \text{ has overflow} \\ h_e^k, & \text{otherwise} \end{cases} \quad (4)$$

with h_{inc} being a constant.

3.2 Motivating Examples

The D-Router starts with any initial routing solution. Particularly, we assume that our initial solution is obtained by the algorithm Liu et al. [2008], but without the congestion constraint, which is actually generated by the first iteration without any congestion constraint in the algorithm Liu et al. [2008]. The D-Router then finds out each highly congested area and spreads out net wires to its neighbors for congestion reduction. The spreading method is derived from the duality with dopant diffusion.

The example in Figure 6 illustrates why the D-Router is free of the routing order, but the algorithm Liu et al. [2008] is not. The traditional substrate routing methods based on A* and the maze cannot handle the cases shown in Figure 6 and Figure 8. A common solder ball assignment case is illustrated in Figure 6(a), where nets A, B, C, and D are routed with a rectangular blockage constraint. If the routing order is D-C-B-A, both algorithms can get the routing result shown in Figure 6(b). However, supposing that we route net A first, then the algorithm Liu et al. [2008] achieves the routing result as Figure 6(c) shows at the earlier iterations with loosening congestion constraint. The iterative scheme in Liu et al. [2008] rips up and reroutes one net at a time, and every

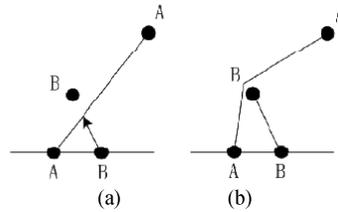


Fig. 7. An example of “dynamic pushing” in routing two nets: (a) routed net A blocks the shortest connection of net B; (b) net B pushes net A for the optimal solution.

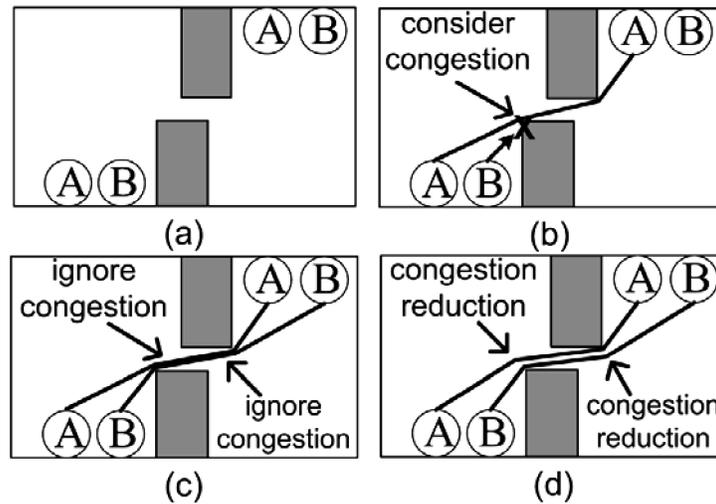
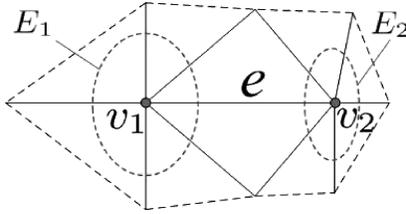


Fig. 8. An example without a valid net ordering.

net is rerouted in each iteration. Thus, due to the congestion constraint, the congested area between the upper-left of the blockage and net A blocks nets B, C, and D from routing through, shown in Figure 6(d). The “dynamic pushing” in Liu et al. [2008] only pushes the blocking net wires, and does not “squeeze” [Staepelaere et al. 1993] through the congested area. Figure 7(a) illustrates that net A blocks the shortest connection of net B, and Figure 7(b) is the routing result with net B dynamically pushing net A. Thus, among those nets A, B, C, and D, only one routing order could achieve the results in Figure 6(b). In practice, identifying such a routing and rerouting order is not inexpensive. However, D-Router reduces congestion from the initial routing in Figure 6(c) by diffusing net A in the congested area to its uncongested neighbor, and the same optimal solution is obtained as Figure 6(b) shows. Hence, D-Router can alleviate the net ordering problem and get more nets routed.

The routing order problem can become more difficult even in a two-net case. Figure 8(a) gives a routing puzzle for the algorithm Liu et al. [2008]. Since the congested area at either the upper-left of the lower blockage, or the lower-right of the upper one, blocks net B or net A from routing through, there is no valid order for routing and rerouting nets A and B, illustrated in Figure 8(b). However, the D-Router can solve it without extra effort by starting from the

Fig. 9. A diffusion window for edge e .

initial solution in Figure 8(c) and diffusing nets A and B in the congested areas to their uncongested neighbors. Finally, the optimal solution in Figure 8(d) is achieved.

4. PLANAR DIFFUSION

4.1 Diffusion Process

Diffusion happens inside an isolated material. The physical process of dopant diffusion is driven by the concentration gradient, which is the slope and steepness of the concentration difference at a given point. It reaches equilibrium when the material concentration is evenly distributed. The commonly used model of diffusion is *continuous, global, and simultaneous*. Our algorithm, however, is discrete, localized, and works net-by-net. In order to simulate the process of dopant diffusion for congestion reduction in the routing phase, given moment t , we define the *concentration* $d_e(t)$ of PCDT edge e as congestion on edge e for moment t .

$$d_e(t) = \eta_e(t) \quad (5)$$

Next, we will define key concepts such as the diffusion window and show that our diffusion is discrete and localized.

4.1.1 Diffusion Window. Congestion caused by net segments and net endpoints is discrete according to values w_i and s_i in Eq. (1). We use the concept of the *diffusion window* as an isolated area for congestion reduction. Given a highly congested PCDT edge e as a *diffusion source*, the diffusion window includes edge e itself and adjacent edges sets E_1 and E_2 . Edges in E_1 and E_2 are incident to vertices v_1 and v_2 of edge e , respectively. The diffusion window is shown with solid lines in Figure 9.

Once a *diffusion window* is created, it is isolated from the outside: that is, there is no congestion being diffused between the outside and inside of the window until the window is released. In the diffusion window, the diffusion is in a one-dimensional (1D) space with two directions towards E_1 and E_2 . In order to guarantee edge e is the diffusion source in a *diffusion window*, the most congested edge (i.e., with the largest $\eta_e(t)$) is chosen in every iteration.

4.1.2 Diffusion Concentration Inside the Window. To solve the discrete routing problem, we discretize the diffusion process within a diffusion window into finite atomic movements of a net segment or a net end-point towards E_1

or E_2 (i.e., away from the diffusion source e). Then, we develop a practical concentration formula based on Eq. (5). When a net moves towards set E_1 or E_2 , it may pass through more than one edge in set E_1 or E_2 . To decide the direction of the diffusion, we first define the *diffused edges* as the edges in E_1 or E_2 through which the net passes. We then consider either side of the diffused edges as a grouped edge and denote it as Edf . Then, we calculate the concentration value of Edf , $d_{Edf}(t)$, by using the maximal congestion of the edge among diffused edges as follows.

$$d_{Edf}(t) = \max_{e_i \in Edf} \{\eta_{e_i}(t)\} \quad (6)$$

4.1.3 Diffusion Velocity. The congestion reduction is formulated as a localized congestion diffusion problem. For position x , a general definition of 1D *diffusion velocity* $v_x(t)$ is as follows.

$$v_x(t) = -\frac{d(d_e(t))}{dx} / d_e(t) \quad (7)$$

Then, from Eqs. (7) and (6), we have the discrete form of velocity for a diffusion source e as

$$v_{e^+}(t) = -(d_{Edf^+}(t) - d_e(t)) / d_e(t), \quad (8)$$

$$v_{e^-}(t) = -(d_{Edf^-}(t) - d_e(t)) / d_e(t), \quad (9)$$

where Edf^+ and Edf^- are the diffused edges in E_1 and E_2 , respectively, and we assume that the moving distance of the atomic movement is 1.

Eqs. (8) and (9) indicate that the diffusion towards the diffused edges with less concentration has a higher speed. Therefore, in the discrete case, we always select the direction with the higher speed to perform the diffusion, which means lower concentration and lower congestion.

4.2 Algorithm Overview

The algorithm overview of the D-Router is given in Figure 10, and key algorithm components are discussed in Sections 4.3 through 4.5. In addition to the procedure of local diffusion in the diffusion window, a heap H and a taboo list Tb are used for the convergence of local diffusion iteration, which will be discussed in Section 4.5. A variable congestion threshold is used to solve the congestion accumulating problem, which is also discussed in Section 4.5. ψ decreases from ψ^* until there is no congested area left. A lower bound of the threshold ψ_0 (can be 0) is used to terminate the iteration while meeting irremovable congestion. Decreasing step $\Delta\psi$ can be assigned as $\psi^*/25$.

Figure 11 presents topological routing before and after diffusion-driven congestion reduction. Diffusion evenly spreads out congested nets while keeping the original topology of nets.

4.3 Momentary-Diffusion Operations

Momentary-diffusion is the atomic net segment/end-point movement from a diffusion source to a selected diffusion direction in the routing problem. We

```

INPUT
Initial routing solution,
PCDT construction on SRG graph,
The given congestion threshold  $\psi^*$ ,
A lower bound of congestion threshold  $\psi_0$ ,
The congestion threshold decreasing step  $\Delta\psi$ 
FOR the threshold  $\psi$  decreases from  $\psi^*$  to  $\psi_0$  by step  $\Delta\psi$ ,
DO
{
IF(no congested edges determined by the threshold  $\psi^*$ ) Break;
Create a heap  $H$  and a taboo list  $Tb$ ;
Insert congested edges determined by the threshold  $\psi$  into  $H$ ;
FOR the most highly congested edge  $e$  from  $H$ ,
DO
{
Find the diffusion window of  $e$ ;
WHILE(!(diffusion window reaches equilibrium) )
{
Choose the diffusion direction with the higher speed
at the moment  $t$ ;
DO momentary-diffusion  $D(n, e, E)$ ;
Update the congestion value  $\eta_{ei}(t)$ ,  $e_i \in Edf$ ;
Update the taboo list  $Tb$  and the heap  $H$ ;
}
Release the diffusion window;
IF( $e$  is still congested) put  $e$  into  $Tb$ ;
Delete  $e$  from  $H$ ;
Re-heapify  $H$ ;
}
}
}

```

Fig. 10. The algorithm of D-Router.

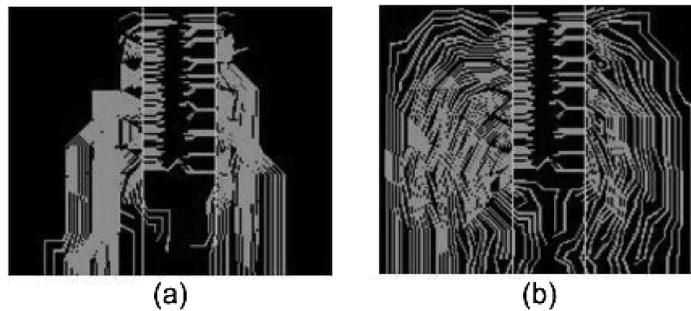


Fig. 11. Routing (a) before and (b) after D-Router.

define the momentary-diffusion operations as $D(n, e, E)$, where n is the net segment to be moved, e is the diffusion source, and E is the diffusion direction such as E_1 or E_2 in Figure 9 which is chosen in the moment. Let vertex v , such as v_1 or v_2 in Figure 9, be the common vertex incident to edge e and edges set E_1 or E_2 . Net segment n should be the one closest to vertex v . Next, we describe the

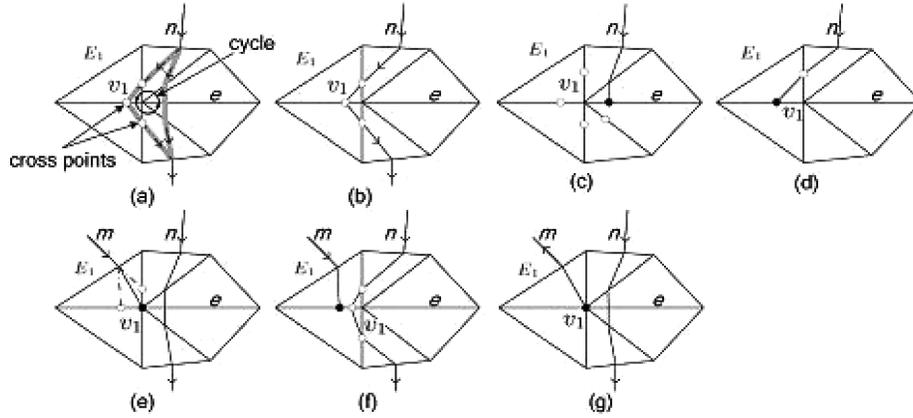


Fig. 12. Momentary-diffusion operations. (a) (*cycle canceling*) and (b) for case 1; (c) and (d) for case 2; (e) and (f) for case 3; (g) for case 4.

momentary-diffusion operations in each case. For simplicity of presentation, we take direction E_1 and vertex v_1 as an example.

Case 1: Normal. In this case, net n moves through diffusion source e ; no net starts from or stops at vertex v_1 and no edge in E_1 is the boundary of a blockage.

The *cycle canceling* method from network flow optimization can be used to maintain a feasible routing solution in every iteration. A net in PCDT can be regarded as a flow from start point to the end-zone. Therefore, the *cycle canceling* method is used to change the segment path. As shown in Figure 12(a), a counter clockwise cycle is found with respect to the direction and location of the net flow. Afterwards, crossing points are added to the closest positions to vertex v_1 in order to keep the original topology of all nets. Then, crossing points are stitched sequentially to form a chain as shown in Figure 12(b). Finally, a new routing path of net n is found and the congestion of edge e is reduced.

Case 2: Net n Stops at a Diffusion Source e . In this case, net n stops at diffusion source e with a via as shown in Figure 12(c).

According to our data structure, when a net stops on a triangle edge, this net must have a flexible end-point and arrive at the end-zone illustrated in Figure 1. After adding crossing points on diffused edges the same way as case 1, we search these crossing points sequentially until we find a crossing point located inside the end-zone with radius R_1 . The searched crossing points are stitched back to the net, forming a new path. Thus, after deleting the redundant crossing points, a new routing result with reduced congestion for net n is obtained in Figure 12(d).

Case 3: Net m Stops at Vertex v_1 Beside Net n . According to the data structure PCDT, flexible end-points can be located at vertices as shown in Figure 12(e). In this case, the end-point of net m is released to the less-congested edge. This release does not change the topology of net m and a new solution of net n with reduced congestion can be easily obtained by the cycle canceling method in case 1, as illustrated in Figure 12(f).

Case 4: Net m Starts from Vertex v_1 Beside Net n . According to the data structure PCDT, a start-point must be located at a triangle vertex. Thus, if vertex v_1 is the start-point of net m , net n cannot move to set E_1 as shown in Figure 12(g). In this case, we should try the other side, such as E_2 , for the diffusion. If the same case appears in the other side, it is usually the result of improper solder ball assignment or improper start-point locations decided by escape routing. Such information can be feedback for the escape routing stage. In addition, keeping initial topological ordering of the routed nets may also cause irremovable congestion. In this case, to tackle the problem, net n can be ripped up completely and rerouted by the method in Liu et al. [2008].

4.4 Diffusion Equilibrium

Any of the following three conditions indicates equilibrium.

Condition I. For a given threshold ψ^* , if congestion $\eta_e(t) < \psi^*$, diffusion reaches equilibrium.

Condition II. The congestion values of every edge in a diffusion window are not necessarily equal to each other when equilibrium is reached. *Overdiffusion* is a momentary-diffusion that makes the diffusion source less congested than an edge in set Edf . Localized diffusion reaches equilibrium when the next momentary-diffusion is overdiffused, so diffusion repeated between two edges back and forth can be avoided.

Condition III. When diffusion source edge e has one vertex v that is the start-point of a net, the diffusion towards this direction cannot be performed. Additionally, when any diffused edge in Edf is the boundary of a blockage, or is forbidden by the taboo list (see Figure 10 and detailed discussion in next subsection), the diffusion towards this direction cannot be made. When diffusion cannot be made towards both directions, diffusion reaches equilibrium.

4.5 Iteration Convergence

A heap H and a taboo list Tb are maintained for the process of diffusion. H maintains all possible diffusion sources and is heapified by edge congestion. Once the heap H is built, the first element inside it is the most highly congested edge. Taboo list Tb maintains all the edges that are no longer allowed to diffuse congestion. When one of the diffused edges Edf is such an edge, momentary-diffusion towards this direction is forbidden. In the beginning, H contains all congested edges as possible diffusion sources and Tb is initially empty. A diffusion source is added into Tb if it is still congested when the localized diffusion reaches equilibrium. An edge in Tb can be released when the concentration of one of its neighbor edges becomes reduced. Every diffusion source is removed from H once its diffusion window reaches equilibrium. Meanwhile, both newly congested and newly released edges are added into H after every momentary-diffusion. Once H is empty, the planar diffusion process terminates.

Our diffusion method is based on discrete concentration and a localized diffusion window. Therefore, if several neighboring diffusion windows reach

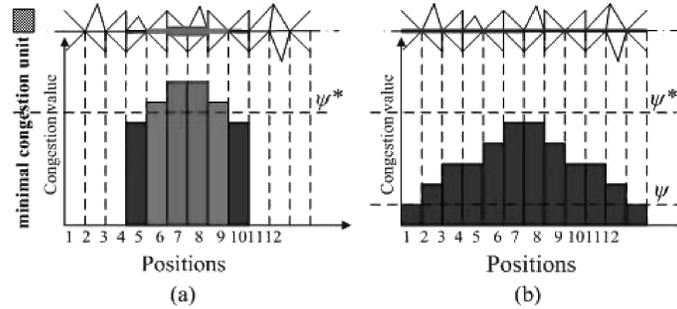


Fig. 13. Congestion accumulation effect and improvement.

equilibrium in Condition II, the congestion on these neighboring edges accumulate, as in positions 4 through 6 in Figure 13(a). Meanwhile, edges as in position 4 in Figure 13(a) with congestion less than ψ^* are not diffused according to Condition I. Thus, congestion easily accumulates more than ψ^* but cannot be reduced, as positions 5 through 8 in Figure 13(a). To solve this, we give a lower congestion threshold ψ as shown in Figure 13(b). Since ψ is smaller, the diffusion in the diffusion window of positions 4 and 9 should be continued (Condition I is unsatisfied). Some net segments in positions 4 and 9 are moved to position 3 and 10, resulting in a larger concentration gradient between positions 4 through 9. This facilitates diffusions in other windows. Then, congestion can be further reduced and the final result through iterative diffusion is shown in Figure 13(b), which spreads routed nets more evenly.

The wire lengths can increase unnecessarily because the diffusion method does not have the flexibility to change the topological ordering of the routed nets. Taking Figure 7(a) as an example, there are two kinds of topological orders for both net A and B: net A detouring net B, and net B detouring net A. When “dynamic pushing” chooses net A detouring net B as shown in Figure 7(b), if such detouring makes net A enter a congested area, diffusion may make net A detour much further away from the congested area. This leads to longer wire length compared to the other topological order where net B detours net A. A lower bound of the congestion threshold ψ_0 , which is used to terminate the iteration, as well as a decreasing step $\Delta\psi$, can help maintain control over the net wire lengths. Additionally, the equilibrium Condition II for the overdifffusion is a safeguard of sorts for the increasing wire length. Meanwhile, without loss of efficiency, another rerouting iteration with congestion constraint Liu et al. [2008] as a postoptimization can reduce the unnecessarily increased wire lengths.

4.6 Complexity Analysis

Let the number of total nets be n . The number of obstacles is relatively minor when compared with n . Also, one cluster of small obstacles can be merged into one obstacle. Thus, the size of the congestion reduction problem can be viewed as n .

In practice, the end-points of nets are distributed uniformly as a grid, and start-points are nearly regularly located. Thus, it is reasonable to assume that

the number of nets passing through a triangle edge is regarded as $O(1)$ compared with n .

In the first FOR-loop of algorithm, the outermost iteration for threshold decrease is a constant $(\psi^* - \psi_0)/\Delta\psi$.

Let the number of total triangles be n_t . Thus, the total triangle edges are $O(n_t)$. Then, the complexity of building the heap H with at most $O(n_t)$ elements is $O(n_t)$.

With the assumption that the number of nets passing through a triangle edge is $O(1)$, the diffusion can reach equilibrium after at most $O(1)$ iterations of momentary-diffusion.

During a localized diffusion, the number of edges in a diffusion window is also constant. Therefore, the complexity of momentary-diffusion operations is $O(1)$.

From the initial solution to the end of congestion reduction, the times of diffusion operations occurring on each triangle edge are less than the number of nets passing through it, $O(1)$. For the majority of triangle edges, at least one net segment is moved by a diffusion operation every time the edge is inserted to and removed from heap H by means of taboo list Tb . Therefore, the total times that an edge is inserted into and removed from heap H and taboo list Tb are $O(1)$.

When removing the first element from the heap and updating congestion, the complexity is $O(\log(n_t))$. Thus, the complexity of a localized diffusion is $O(\log(n_t))$ and the complexity of the algorithm is $O(n_t \log(n_t))$.

Furthermore, data structure PCDT uses the BGA solder ball grid pattern as the size of inserted particles while the number of obstacles can be neglected compared with that of solder balls. In an actual design case, the number of solder balls in the BGA layer should be $O(n)$. The number of total elements in PCDT is $(O(n)+2O(n))$ considering particles, obstacles, and start-/end-points of nets. Therefore, $n_t = O(n)$.

The complexity of the algorithm in Figure 10 is $O(n \log(n))$ with the assumption that the number of nets passing through a triangle edge is $O(1)$, where n is the number of total net.

5. VERTICAL DIFFUSION

Diffusion can be performed vertically between layers. The destination of each two-pin net is the solder ball in the bottom layer, and lower routing layers are always designed to be less congested. Thus, the diffusion window for vertical diffusion may encompass an arbitrary number of layers down to the next build-up layer for signal routing. Diffusion direction always faces downwards by dropping vias to avoid vertical detour that is forbidden by design rules.

Vertical diffusion starts where a failed net stops during planar diffusion. If the location is uncongested for creating a via, the net diffuses to the next signal routing layer by staggered or stacked vias according to the given via technology.² Usually, vertical diffusion goes through a couple of core layers, or

²There are usually a limited number of failed nets after planar diffusion. Thus we can often find uncongested locations for creating vias, which do not conflict with the end-zone constraint.

Table I. Test Case Characters

Test case ID	Package type	Package size*	Die(s) size*	Number of nets
Q1	2-0-2	10000×10000	7500×7700	315
B2	2-2-2	35000×35000	14000×15000	474
F3	2-2-2	30000×30000	9000×10500	543
P4	3-1-3	40000×40000	9300×9300	800
A5	3-2-3	35000×35000	12000×12000	506
A6	3-2-3	40000×40000	20000×22000	891
X7	4-2-4	40000×40000	20000×23000	990
A8	4-2-4	45000×45000	20000×19000	1009
S9	1-0-1	12000×12000	3900×6700 4400×5700 3200×4400	349
S10	2-2-2	37500×37500	11000×10000 4700×3800 4600×5500	538
total	—	—	—	6415

(*: Package size and Die(s) size are given by width × length (μm) in rectangle.)

build-up layers, until it reaches the next signal routing layer. Therefore, when via is staggered, the direction towards the end-point of the net is preferred.

Vertical diffusion can be embedded inside planar diffusion, where a failed net can only stop at a triangle edge because of the data structure PCDT. During momentary-diffusion, a vertical diffusion will be considered the same as case 2 in operations $D(n, e, E)$. In this case, the failed net stops at an uncongested or less-congested edge of PCDT. In the lower routing layer, the failed routed nets from the one above will route initially and reduce congestion together with other nets starting from this layer.

6. EXPERIMENTAL RESULTS

The proposed algorithm D-Router has been implemented in the C++ language and integrated into an industrial package design tool-set for substrate topology routing.

Table I summarizes the test case characteristics, including the package type and size, die size, and total number of nets. The package type indicates the number of build-up layers and core layers. The package substrate always keeps a symmetric structure, for the balance of thermal dilation. Thus, a type of m - n - m substrate indicates that there are n core layers and $2m$ build-up layers. All the core layers and some of the build-up layers are used for P/G planes, while the other build-up layers are for signal routing, that is, substrate routing. The first eight test cases are single-chip packages while the remaining two have multiple dies within the packages. The last nine test cases are from Liu et al. [2008] that have already given fairly decent solutions, as does the D-Router. However, designers practically prefer some I/Os to connect to solder balls in specified regions for the sake of PCB design. In our experiments, region constraints are considered in the solder ball assignment and the solder balls are reassigned

Table II. Experimental Results

Test case	Number of failed nets				σ : standard deviation of congestion			Wire length (mm)			Runtime (s)		
	[Liu's]	Nego	D-R		[Liu's]	Nego	D-R	[Liu's]	Nego	D-R	[Liu's]	Nego	D-R
			P	P+V									
Q1	51	41	33	26	0.45	0.34	0.25	1.64	1.69	1.70	5.34	9.79	7.17
B2	31	30	0	0	0.21	0.21	0.15	6.98	6.98	7.17	11.39	17.84	9.00
F3	24	22	0	0	0.18	0.17	0.15	7.79	7.80	7.96	14.36	14.41	16.91
P4	135	135	55	48	0.33	0.33	0.24	11.90	11.90	12.30	41.64	20.64	13.87
A5	64	63	7	7	0.16	0.16	0.15	14.90	14.90	16.30	13.27	17.65	10.92
A8	60	57	17	15	0.19	0.19	0.16	4.98	4.99	4.93	12.12	18.77	12.87
X7	45	45	8	8	0.22	0.22	0.19	6.55	6.54	6.53	39.51	45.11	25.38
A8	16	16	0	0	0.21	0.21	0.20	18.50	18.50	18.70	44.55	47.24	9.32
S9	22	20	0	0	0.18	0.18	0.14	1.67	1.67	1.85	2.11	3.2	0.96
S10	32	32	0	0	0.21	0.21	0.19	9.53	9.53	7.90	284.17	286.34	3.01
total	480	461	120 1/4.0x 1/3.8x	104 1/4.6x 1/4.4x	—	—	—	—	—	—	—	—	—
average	—	—	—	—	0.23	0.22	0.18 -21.7% -18.2%	8.45	8.46	8.51	46.05	47.04	10.94 1/4.2x 1/4.3x

(Nego: a negotiation-based congestion reduction mechanism is used in the ripping-up-and-rerouting procedure in [Liu et al. 2008].)

(P: the D-Router just uses planar diffusion.)

(P+V: the D-Router uses both planar diffusion and vertical diffusion.)

([Liu's]: the algorithm proposed in Liu et al. [2008].)

(D-R: the D-Router proposed in this article.)

before substrate routing without a loss of reality. The netlist is changed, which becomes harder to route by Liu et al. [2008]. The harder routing problems are illustrated in Figure 6 and 8. Furthermore, we gave new names to the nine test cases in order to distinguish from those in Liu et al. [2008]. The experiments occurred in a Linux-2.6 server with 2.4 GHz dual CPUs and 2GB memory.

Table II compares the D-Router with Liu et al. [2008] and the negotiation-based substrate routing (called “Nego” in Table II) introduced in Section 3. We do not compare our results with Surf since problem formulations are different and the effort to reimplement Surf for our problem formulation is hardly justified. The D-Router with both planar and vertical diffusion has 104 failed nets while the other two algorithms have 480 and 461 failed nets, respectively. Most failed nets in the compared algorithms are fixed by planar diffusion and about 13% ((120–104)/120) failed nets are fixed by vertical diffusion. σ is the standard deviation of the congestion map, and is calculated as

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (\eta_i - \bar{\eta})^2}, \quad (10)$$

where N is the total number of PCDT edges through which there exists a net passing, η_i is the final congestion value of edge e_i after diffusion, and $\bar{\eta}$ is the mean of η_i . A smaller σ indicates that the wires are spreading out more evenly. The D-Router reduces the standard deviation σ of congestion by up to 21.7%, which means our algorithm achieves a more even routing solution.

We also measure wire length calculated for all nets that can be routed by all three algorithms and the three algorithms obtained similar length. However, the D-Router actually reduced wire length slightly after manually routing all nets due to its higher routability. The runtime of the initial routing of the D-Router is also added into the runtime as a whole, because in this way, the D-Router can be viewed as an identical functional module for comparison with the two alternative algorithms. Additionally, the initial routing actually takes only one iteration of the algorithm Liu et al. [2008] without any congestion

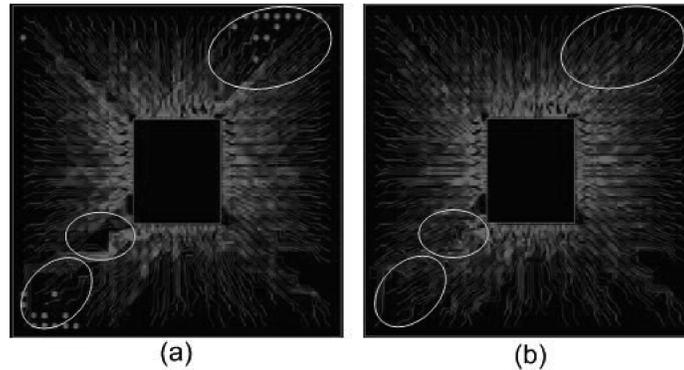


Fig. 14. Comparison of (a) Liu et al. [2008] and (b) D-Router.

constraint. Liu et al. [2008] and negotiation-based algorithms have similar runtimes, while the D-Router reduces runtime by on average 4.3 times but up to 94 times compared with the two alternative algorithms for the large-scale example S10. Test case S10 is a system-in-package package with multiple dies in one package; there are more complex routing problems shown in Figure 8. As for the two alternative algorithms, such a problem requires more dynamic searching steps [Liu et al. 2008] and even more when a net fails to be connected so that they search all the possible triangles. Furthermore, more iterations are also required for convergence. However, the initial routing of D-Router doesn't consider congestion constraints, thus a routing solution is easily found. The congestion diffusion process has the same complexity with other cases. Therefore, both alternative algorithms have much longer runtimes than the D-Router, especially for test case S10. This demonstrates the quality of D-Router most effectively.

In order to give a direct view of the performance of the D-Router, Figure 14(a) and 14(b) show routing results of one whole layer of case F3, which are obtained by Liu et al. [2008] and the D-Router, respectively. In addition, two magnified views of a corner of cases B2 and X7 are shown for both algorithms in Figure 15(a), 15(b) and Figure 15(c) and 15(d), respectively, in which 15(a) and 15(b) are the results of Liu et al. [2008], and 15(c) and 15(d) are that of the D-Router. As these figures show, the highlighted dots are the end-points of uncompleted nets. The highlighted circles show the congestion reduction.

Therefore, our algorithm D-Router achieves superior routing results, a higher ratio of completely routed to failed nets, and less runtime without increasing average wire length.

7. CONCLUSIONS

Off-chip substrate routing for high-density packages is challenging and existing substrate routing algorithms often result in a large number of unrouted nets that have to be routed manually. We have developed a diffusion-based topological router (D-Router) for congestion reduction in substrate routing. It

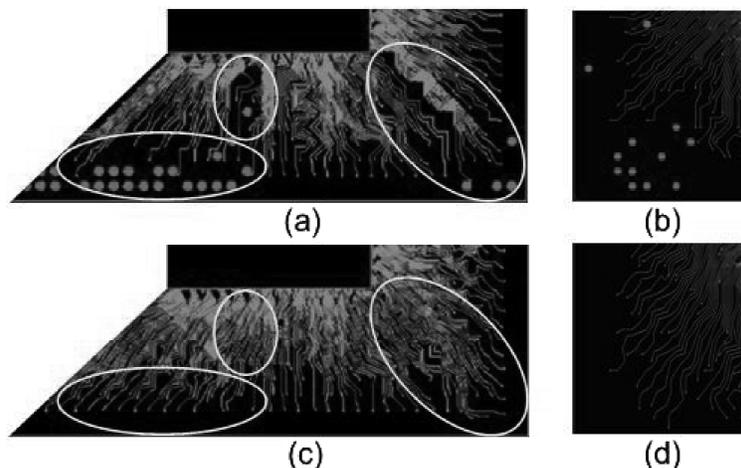


Fig. 15. Comparison of magnified view of the corner of cases B2 and X7.

improves routability by a simulated diffusion process based on the duality between congestion and concentration. Experiments using industrial design examples show that a very recent substrate routing method [Liu et al. 2008] leaves 480 nets unrouted for ten industrial designs with a total of 6415 nets, while the D-Router reduces the unrouted nets to 104, a 4.6 times net number reduction, which translates to substantial design time reduction. Our algorithm also reduces runtime by on average 4.3 times but up to 94 times.

While the negotiation-based routing algorithm obtains good on-chip routing results for both FPGA and ASIC [McMurchie and Ebeling 1995; Roy and Markov 2007; Cho et al. 2007], results in this article show that the D-Router significantly outperforms the negotiation-based algorithm in terms of both routability and runtime in the package substrate routing problem. Although the D-Router is used for substrate routing in this article, its concept can be applied to on-chip routing as well. Extending the D-Router to on-chip routing will be our future work.

REFERENCES

- BOSSON, F. 1996. Anisotropic mesh generation with particles. M.S. thesis, Carnegie Mellon University.
- CHEN, H. F. S. AND LEE, D. T. 1996. A faster algorithm for rubber-band equivalent transformation floorplanar VLSI layouts. *IEEE Trans. Comput.-Aid. Des. Integr. Circ. Syst.* 15, 2, 217–227.
- CHO, M., LU, K., YUAN, K., AND PAN, D. 2007. BoxRouter 2.0: Architecture and implementation of a hybrid and robust global router. In *Proceedings of the International Conference on Computer-Aided Design*. 503–508.
- CHO, M. AND PAN, D. 2006. BoxRouter: A new global router based on box expansion and progressive ILP. In *Proceedings of the IEEE/ACM Design Automation Conference (DAC'06)*. 373–378.
- DAI, W. W., DAYAN, T., AND STAEPELAERE, D. 1991a. Topological routing in SURF: Generating a rubber-band sketch. In *Proceedings of the IEEE/ACM Design Automation Conference (DAC'91)*. 39–44.
- DAI, W. W., KONG, R., AND SATO, M. 1991b. Routability of a rubber-band sketch. In *Proceedings of the IEEE/ACM Design Automation Conference (DAC'91)*. 45–48.

- FANG, J. W., HSU, C. H., AND CHANG, Y. W. 2007. An integer linear programming based routing algorithm for flip-chip design. In *Proceedings of the IEEE/ACM Design Automation Conference (DAC'07)*. 606–611.
- FANG, J. W., LIN, I. J., CHANG, Y. W., AND WANG, J. H. 2005. A routing algorithm for flip-chip design. In *Proceedings of the International Conference on Computer-Aid. Design*. 753–758.
- HORIUCHI, M., YODA, E., AND TAKEUCHI, Y. 2000. Escape routing design to reduce the number of layers in area array packaging. *IEEE Trans. Comput.-Aid. Des. Integr. Circ. Syst.* 23, 4, 686–691.
- KUBO, Y. AND TAKAHASHI, A. 2005. A global routing method for 2-layer ball grid array packages. In *Proceedings of the International Symposium on Physical Design*. 36–43.
- LIU, S. H., CHEN, G. Q., JING, T. T., HE, L., ZHANG, T. P., DUTTA, R., AND HONG, X. L. 2008. Topological routing to maximize routability for package substrate. In *Proceedings of the IEEE/ACM Design Automation Conference (DAC'08)*. 566–569.
- LIU, S. H., CHEN, G. Q., JING, T. T., HE, L., DUTTA, R., AND HONG, X. L. 2009. Diffusion-Driven congestion reduction for substrate topological routing. In *Proceedings of the International Symposium on Physical Design*.
- MCMURCHIE, L. AND EBELING, C. 1995. PathFinder: A negotiation-based performance-driven router for FPGAs. In *Proceedings of the ACM International Symposium on Field-Programmable Gate Arrays*. 111–117.
- OZDAL, M. M. AND WONG, D. F. 2002. Simultaneous escape routing and layer assignment for dense PCBs. In *Proceedings of the International Conference on Computer-Aided Design*. 822–829.
- OZDAL, M. M., WONG, M. D. F., AND HONSINGER, P. S. 1995. An escape routing framework for dense boards with high-speed design constraints. In *Proceedings of the International Conference on Computer-Aided Design*. 759–766.
- PASCARIU, G., CRONIN, P., AND CROWLEY, D. 2003. Next generation electronics packaging utilizing flip chip technology. In *Proceedings of the IEEE Electronics Manufacturing Technology Symposium*. 423–426.
- REN, H., PAN, D. Z., ALPERT, C. J., AND VILLARRUBIA, P. 2005. Diffusion-Based placement migration. In *Proceedings of the IEEE/ACM Design Automation Conference (DAC'05)*. 515–520.
- ROY, J. AND MARKOV, I. 2007. High-Performance routing at the nanometer scale. In *Proceedings of the International Conference on Computer-Aided Design*. 496–502.
- SHI, R. AND CHENG, C. K. 2006. Efficient escape routing for hexagonal array of high density I/Os. In *Proceedings of the IEEE/ACM Design Automation Conference (DAC'06)*. 1003–1008.
- STAEPELAERE, D., JUE, J., DAYAN, T., AND DAI, W. W. 1993. SURF: A rubber-band routing system for multichip modules. *IEEE Trans. Des. Test Comput.* 10, 4, 18–26.
- TSAI, C. C., WANG, C. M., AND CHEN, S. J. 1998. NEWS: A net-even-wiring system for the routing on a multilayer PGA package. *IEEE Trans. Comput.-Aid. Des. Integr. Circ. Syst.* 17, 2, 182–189.
- WANG, R. S., SHI, R., AND CHENG, C. K. 2006. Layer minimization of escape routing in area array packaging. In *Proceedings of the International Conference on Computer-Aided Design*. 815–819.
- WINKLER, E. 1996. Escape routing from chip scale packages. In *Proceedings of the IEEE Electronics Manufacturing Technology Symposium*. 393–401.
- XIONG, J. J., WONG, Y. C., SARITO, E., AND HE, L. 2006. Constraint driven I/O planning and placement for chip- package co-design. In *Proceedings of the Asia South Pacific Design Automation Conference*. 207–212.
- YU, M. F. AND DAI, W. W. 1995. Pin assignment and routing on a single-layer pin grid array. In *Proceedings of the Asia South Pacific Design Automation Conference*. 203–208.
- YU, M. F., DARNAUER, J., AND DAI, W. W. 1996. Interchangeable pin routing with application to package layout. In *Proceedings of the International Conference on Computer-Aided Design*. 668–673.

Received May 2008; revised November 2009; accepted February 2010