Rectilinear Block Placement Using Sequence-Pair

Jin Xu — Cadence Design Systems, Inc., San Jose, CA 95134, jinxu@cadence.com

Pei-ning Guo — University of California, San Diego, La Jolla, CA 92093-0114, pnguo@cs.ucsd.edu

Chung-kuan Cheng — University of California, San Diego, La Jolla, CA 92093-0114, kuan@cs.ucsd.edu

Abstract

With the recent advent of deep sub-micron technology and new packaging schemes such as Multi-Chip Modules(MCMs), integrated circuit components are often not rectangular. Most existing block placement approaches, however, only deal with rectangular blocks, resulting in inefficient area utilization. New approaches which can handle arbitrarily shaped blocks are essential to achieve high performance design. In this paper, we present an approach extending the sequence-pair approach for rectangular block placement to arbitrarily sized and shaped rectilinear blocks. Experimental results show that our algorithm achieves results with excellent area utilization.

1. Introduction

With the recent advent of deep sub-micron technology and new packaging schemes such as Multi-Chip Modules(MCMs), integrated circuit components are often not rectangular. Most existing block placement approaches, however, only deal with rectangular blocks, resulting in inefficient area utilization. New approaches which can handle arbitrarily shaped blocks are essential to achieve high performance design.

Kang et al.[3] proposes a genetic simulated annealing algorithm for L-shaped, T-shaped and soft blocks. Based on Bounded-Slicing Structure([4]), the algorithm combines the SA-based local search and GA-based global crossover for general non-slicing floorplanning.

Lee[5] extends the zone refinement technique introduced by Shin et al.[8] to arbitrarily shaped rectilinear and soft blocks. A rectilinear block is represented by four linear profiles viewed from four directions. A profile is specified by a series of line segments, each of which is defined by two breaking points. A bounded 2D contour searching algorithm is proposed to find the best position for the block.

Preas et al.[10] proposes a graph model for the topological relationship between rectangular blocks. An iterative improvement algorithm is presented to reduce both area and interconnections. Wong el al.[9] extends the Polish expression to represent floorplans of rectangular and L-shaped blocks. A simulated annealing method is used to search for optimal floorplan.

Murata et al.[1] proposes the sequence-pair approach for rectangular block placement. The general idea is to first place the blocks on a grid, and then use the longest path algorithm to estimate the area required by the corresponding compacted placement. To determine the block placement, two block name sequences are derived, which correspond to the horizontal and vertical grid lines. Then they introduce a P-admissible solution space of size $(n!)^2 8^n$, where n is the total number of blocks, and apply a simulated annealing method to search for a good solution.

In this paper, we extend the sequence-pair approach described above to arbitrarily sized and shaped rectilinear blocks. The major contribution of our work is to identify feasible vs. infeasible sequence-pairs for rectilinear blocks. First, we explore the properties of L-shaped blocks, then decompose arbitrarily shaped rectilinear blocks into a set of sub-L-shaped-blocks. The properties of L-shaped blocks, therefore, can be applied to general rectilinear blocks.

To demonstrate the efficiency of our algorithm, we apply it to a randomly generated test case and the modified MCNC benchmark circuit *ami49*. The experiment results show that the algorithm achieves placements with excellent area utilization.

2. Preliminaries -- Sequence-pair

The topological relationships between rectangular blocks can be expressed by two sequences of the block names. And given the dimensions of each block, a placement can be generated by a compaction operation. For more details, please refer to [1].

3. Rectilinear Block Placement

A rectilinear block placement is feasible iff

(1) it is non-overlapping;

(2) all rectilinear blocks are in their original shape.

To handle rectilinear blocks, we partition rectilinear blocks into rectangular blocks.

Definition: sub-block

A rectilinear block can be partitioned into a few rectangular blocks. These are called *sub-blocks*, to distinguish them from individual rectangular blocks.

Suppose $a_1, a_2, ..., a_k$ are the sub-blocks of a rectilinear block, the rectilinear block is denoted by $\{a_1, a_2, ..., a_k\}$.

Obviously, if all rectilinear blocks are just placed as rectangular blocks, i.e. by their bounding boxes(Fig. 1(a)), feasible placements always exist. And similar to rectangular blocks, given a feasible placement, there always exists a corresponding sequence-pair([1]).

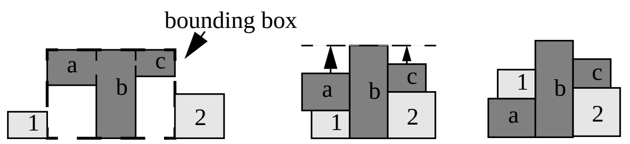
On the other hand, after partitioning and compaction, a rectilinear block might not be in its original shape, as illustrated in Fig. 1(b) and (c). In the case of Fig. 1(b), we can pull sub-blocks *a* and *c* up to align with sub-block *b*, so that the rectilinear block {*a*, *b*, *c*} can maintain its original shape without changing the overall topology of the blocks. There is no room to do so in the case of Fig. 1(c). In other word, a feasible placement can be generated from the sequence-pair in Fig. 1(b) with a local adjustment, and it is, however, impossible from the one in Fig. 1(c).

Now we can define the feasibility of a sequence-pair for rectilinear block placement as follows.

Definition: feasibility of sequence-pair

A sequence-pair is *feasible* if it corresponds to a feasible placement, i.e., after rectangular sub-block compaction, if adjusted locally,
(1) the shapes of rectilinear blocks are maintained,
(2) and the chip size is not changed.

Therefore, the sequence-pair in Fig. 1(b) is feasible, while the one in Fig. 1(c) is not.



$\begin{pmatrix} 1\ a\ b\ c\ 2 \\ 1\ a\ b\ c\ 2 \end{pmatrix}$ $\begin{pmatrix} a\ 1\ b\ c\ 2 \\ 1\ a\ b\ 2\ c \end{pmatrix}$ $\begin{pmatrix} 1\ a\ b\ c\ 2 \\ a\ 1\ b\ 2\ c \end{pmatrix}$

(a) (b) (c)

Fig. 1 (a) Place rectilinear block as a rectangular block; (b) a feasible sequence-pair; (d) an infeasible sequence-pair.

Therefore, the following theorem holds.

Theorem 1:

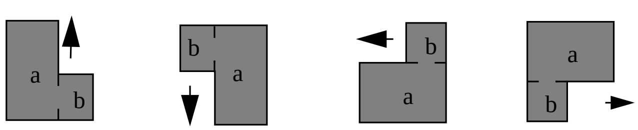
Given a feasible placement, there always exists a corresponding feasible sequence-pair.

In the remaining part of the paper, we will extend the sequence-pair approach to find a feasible sequence-pair which corresponds to a feasible placement with efficient area utilization.

3.1 L-shaped block

Let us consider L-shaped blocks first. In the following discussion we will show that, the feasibility of a placement and the corresponding sequence-pair depends on both the orientations of L-shaped blocks and the relative positions of other blocks in the pair.

Now we can define the orientation of a L-shaped block as follows..



(a) up (b) down (c) left (d) right

Fig. 2 Orientation of a L-shaped block after partitioning

Definition: orientation

After partitioning, a L-shaped block can have four *orientations*, 'up', 'down', 'left' and 'right', as determined by the direction the indented region faces along the partition line, as shown in Fig. 2Corresponding to one orientation, a L-shaped block may have two reflections. Because the same rules apply to the two reflections, only four orientations will be discussed.

To represent the relative positions of a block in a sequence-pair, we can draw directed lines between its two positions in the pair. Fig. 3 shows an example, where *a* and *b* are sub-blocks of a L-shaped block, and *1*, *2* and *3* are rectangular blocks. Blocks *1*, *2* and *3* are between *a* and *b* in at least one of the sequences. We can draw directed lines for each of them, starting from the position which is between *a* and *b*.

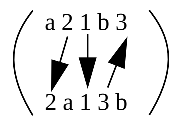


Fig. 3 Relation vectors of blocks

Definition: relation vector

For each block in a sequence-pair, we can draw directed line between its two positions in the two sequences. This line is called *relation vector* of the block.

And any relation vector, if starting from the origin, can be categorized into five classes: I, II, III, IV, and 90^0, according to the quadrant in which the vector ends, as shown in Fig. 4.

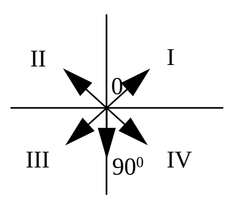


Fig. 4 Classes of relation vectors

Therefore, the relation vectors of blocks *1*, *2* and *3* in Fig. 3 are class 90^0, III, and I respectively. For simplicity, we say that blocks *1*, *2* and *3* are class 90^0, III, and I respectively.

When a L-shaped block {*a*, *b*} is placed with a rectangular block *1*, with respect to the 'up' orientation of {*a*, *b*}, block *1* can be placed anywhere which does not destroy the original shape of {*a*, *b*}. Consequently, the following two restrictions apply:

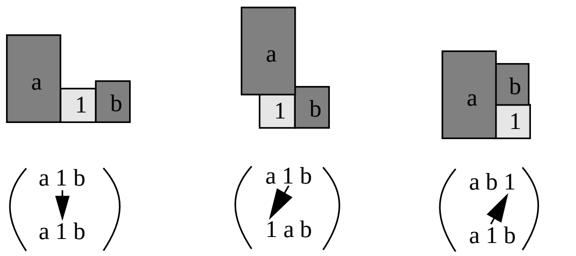
(a) block *1* cannot be class 90^0, as illustrated in Fig. 5(a);

(b) block *1* cannot be class III(Fig. 5(b)), or class I(Fig. 5(c))

All the other sequence-pairs are feasible with respect to this L-shaped block {*a*, *b*}.

Thus, the following lemma holds.

.



(a) class 90^0 (b) class III (c) class I

Fig. 5 Infeasible placements and sequence-pairs with respect to ‘up’ orientation of L-shaped block {*a*, *b*}

Lemma 1:

Given the ‘up’ orientation of a L-shaped block, a sequence-pair is feasible iff all other blocks are either classes II or IV.

Similarly, given the other three orientations, the following lemmas hold.

Lemma 2:

Given the ‘down’ orientation of a L-shaped block, a sequence-pair is feasible iff all other blocks are either classes I or III.

Lemma 3:

Given the ‘left’ orientation of a L-shaped block, a sequence-pair is feasible iff all other blocks are either classes II or III.

Lemma 4:

Given the ‘right’ orientation of a L-shaped block, a sequence-pair is feasible iff all other blocks are either classes I or IV.

Therefore, the following theorem holds.

Theorem 2:

A sequence-pair is feasible with respect to one specific orientation of a L-shaped block iff it satisfies Lemma 1 through 4.

3.2 General rectilinear block

In general, a rectilinear block can be partitioned into a few rectangular sub-blocks. And in order to employ the approach described above, the partitioning has to be done in such a way that two neighbouring sub-blocks can be grouped into a L-shaped block.

Definition: sub-L-shaped-block

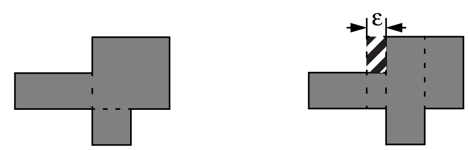
A rectilinear block can be represented as a set of L-shaped blocks. These are called *sub-L-shaped-blocks*, to distinguish them from individual L-shaped blocks.

A rectilinear block can be partitioned in more than one way, vertically or horizontally. To represent it as a set of L-shaped blocks, it is partitioned in only one direction, and all two neighboring sub-blocks should align in the orthogonal direction. Fig. 6 illustrates an example. Only the partitioning in Fig. 6(a) is feasible.

(a) feasible (b) infeasible

Fig. 6 Feasible vs. infeasible partitioning of blocks

Some rectilinear blocks, however, cannot be partitioned into a set of sub-L-shaped blocks, neither vertically nor horizontally, as illustrated in Fig. 7(a). We can select a side of the rectilinear block and expand it by the size of ε, as shown in Fig. 7(b), so that the rectilinear block is L-shape dividable. We call this operation ε-approximation of the rectilinear block.



(a) original (b) ε-approximation

Fig. 7 ε-approximation of rectilinear block

Therefore, the following theorem holds.

Theorem 3:

The sequence-pair approach can be applied to rectilinear block placement iff each rectilinear block can be partitioned into a set of sub-L-shaped blocks with appropriate ε-approximation.

Obviously, these sub-blocks should always maintain their initial relative positions in any feasible placements. Fig. 8 shows all the eight orientations and reflections of a T-shaped block and the corresponding sequence-pairs. As we can see, block *b* is always between block *a* and *c* in all of the placements and pairs. All other sequences, such as “*a c b*” and “*b c a*”, are infeasible. Consequently, there are only eight feasible pairs for the T-shaped block {*a, b, c*}, instead of $(3!)^2 = 36$([1]).

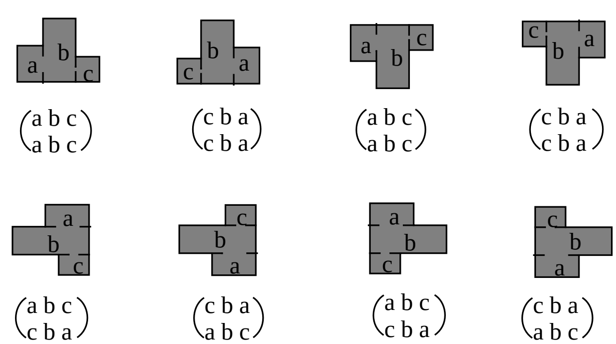


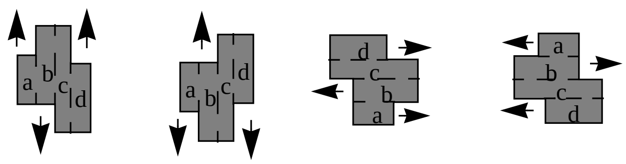
Fig. 8 Feasible sequence-pairs for the sub-blocks of a T-shaped block

In general, we have the following theorem.

Theorem 4:

The number of feasible sequence-pairs for the sub-blocks of a rectilinear block is always only eight, instead of $(k!)^2$, where k is the number of the sub-blocks.

And when the rectilinear block is rotated or reflected, those sub-L-shaped-blocks are also rotated or reflected accordingly, and should always maintain their initial orientation relationships, as illustrated in Fig. 9.



(a) (b) (c) (d)

Fig. 9 Orientations of the sub-L-shaped-blocks (a) original; (b) reflected up/down; (c) rotated 90^0; (d) rotated -90^0

A rectilinear block can be categorized as convex or concave block, according to its shape. The definition is as follows.

Definition: convex

A rectilinear block is *convex* if any two points within the block can be connected by the shortest Manhattan path, which is also within the block. Otherwise the block is *concave*, as illustrated in Fig. 12.

convex concave

Fig. 10 Rectilinear convex vs. concave blocks

Some convex rectilinear blocks can be partitioned into a few sub-L-shaped blocks which are aligned in one side. As shown in Fig. 11(a), an arbitrarily shaped rectilinear block is partitioned into three sub-blocks *a*, *b* and *c*. Those three sub-blocks form two L-shaped blocks {*a*, *b*} and {*b*, *c*}, which orientations are both ‘up’.

(a) mound-shaped (b) Z-shaped

Fig. 11 Partition rectilinear blocks to sub-L-shaped-blocks

The Z-shaped block shown in Fig. 11(b) can be represented by the two sub-L-shaped-blocks {*a*, *b*} and {*b*, *c*}, which orientations are ‘down’ and ‘up’ respectively.

Definition: mound-shaped

If a rectilinear convex block can be partitioned into a few sub-L-shaped blocks with the same orientations, the rectilinear block is called *mound-shaped* rectilinear block.

Therefore, the block in Fig. 11(a) is mound-shaped, while the Z-shaped block in Fig. 11(b) is not.

For mound-shaped rectilinear blocks, all the sub-L-shaped blocks have the same orientation. This property will assure the alignment of all L-shaped blocks in placement. Therefore, the theorem and lemmas presented in Section 3.1 for L-shaped blocks can be applied to mound-shaped rectilinear blocks without any limitations.

For general rectilinear blocks, their sub-L-shaped blocks may have different orientations. This may increase the difficulty of aligning all L-shaped blocks in their original orientations. Fig. 12 illustrates a placement of a Z-shaped block {*a*, *b*, *c*} and two rectangular blocks *1* and *2*. We must expand sub-block *b* in y-direction to maintain the original shape of the Z-shaped block.

(a) before expansion (b) after expansion

Fig. 12 Y-direction expansion of sub-block *b* of Z-shaped block {*a*, *b*, *c*}

Similarly, a concave rectilinear block can also be placed if its sub-blocks are expandible.

In Fig. 13(a), the placement is infeasible, because the concave rectilinear block {*a*, *b*, *c*} cannot maintain its original shape. But if the sub-block *b* is expandible in x-direction, the placement in Fig. 13(b) is feasible.

(a) (b)

Fig. 13 Place concave rectilinear block with expansion

Therefore, the following theorem holds.

Theorem 5:

With respect to an arbitrarily shaped rectilinear block, a sequence-pair is feasible iff it is feasible with respect to all the sub-L-shaped-blocks of the rectilinear block, assuming the sub-blocks are expandible.

4. Algorithm

Similar to Murata’s approach, we also apply a standard simulated annealing strategy to search the solution space. For rectilinear blocks, however, infeasible sequence-pairs might be generated. Consequently, we make the feasibility of sequence-pairs as part of the cost function used in the annealing process. If an infeasible pair is identified, we add a penalty to the cost function according to the number of the infeasible blocks. A carefully selected cooling schedule will converge the annealing process and reach an optimal configuration.

The outline of the algorithm is as follows.

```
Procedure PLACE
begin
    s := initial configuration with random sequence pairs
    T := T0;
    repeat
        count := 0;
        repeat
            count := count + 1;
            nexts := generate(s);
            if cost(nexts) <= cost(s) or
                f(cost(s),cost(nexts),T) > random(0,1)
            then s := nexts;
        until equilibrium(count, s, T);
        T := update(T);
    until the time reaches the limit or Frozen(T);
    output the best placement found;
end

Function Generate(s)
begin
    apply one of the following move operations to perturb
the sequences in s;
        swap two blocks in one of the sequence;
        swap two blocks in both the first and the second
sequences;
        rotate one block;
    return modified s;
end

Function Cost(s)
begin
    C1 := Area_evaluation(s);
    C2 := Wire_lenght_estimation(s);
    C3 := Infeasible_penalty(s);
    return W1*C1 + W2*C2 + W3*C3;
end

Procedure Initialization
Input: rectilinear blocks
Output: a set of sub-L-shaped-blocks
begin
    for each rectilinear block
        partition it into rectangular sub-L-shaped blocks
        employ e-approximation if necessary
        set orientation of each sub-L-shaped block
        store the information of all sub-blocks.
end
```

To show the efficiency of our algorithm, we apply it to a randomly generated test case, and two modified MCNC benchmark circuit *ami49*. The experimental results are shown in the table and placements are in Fig. 14.

5. Conclusions

In this paper, we extend the sequence-pair approach introduced in [1] for rectangular block placement to arbitrarily sized and shaped rectilinear blocks. The properties of L-shaped blocks are examined first, and then arbitrarily shaped rectilinear blocks are decomposed into a set of L-shaped-blocks. The properties of L-shaped blocks, therefore, are applied to general rectilinear blocks. The experiment results show that the algorithm achieves placements with excellent area utilization.

Acknowledgement

This work has been done when the first author was with University of California at San Diego, and was supported in part by grants from the NSF project MIP-9529077 and the California MICRO Program.

Reference

[1] H. Murata, K. Fujiyoshi, S. Nakatake, Y. Kajitani, “Rectangular-Packing-Based Module Placement”, *Proc. IEEE International Conf. on Computer-Aided Design*, pp. 472-479, 1995.

[2] J. Xu, P. N. Guo, C. K. Cheng, “Cluster Refinement for Block Placement”, *Proc. ACM/IEEE Design Automation Conf.*, pp. 762-765, June 1997.

[3] M. Kang, W. W.-M. Dai, “General Floorplanning with L-shaped, T-shaped and Soft Blocks Based on Bounded Slicing Grid Structure”, *Proc. 1997 ASP-DAC*, pp. 265-270, 1997.

[4] S. Nakatake, H. Murata, K. Fujiyoshi, Y. Kajitani, “Bounded-Slicing Structure for Module Placement”, *Technical Report of the Institute of Electronics, Information and Communication Engineers of Japan*, vol. VLD94, pp. 19-24, 1994.

[5] T. C. Lee, “A Bounded 2D Contour Searching Algorithm for Floorplan Design with Arbitrarily Shaped Rectilinear and Soft Modules”, *Proc. 30th Design Automation Conf.*, pp. 525-530, 1993.

[6] C.K. Cheng, E.S. Kuh, “Module Placement based on Resistive Network Optimization”, *IEEE Trans. Computer-Aided Design*, vol. CAD-3, pp. 218-225, July 1984.

[7] H. Onodera, Y. Taniguchi, K. Tamaru, “Branch-and-Bound Placement for Building Block Layout”, *Proc. 28th Design Automation Conf.*, pp. 433-439, 1991.

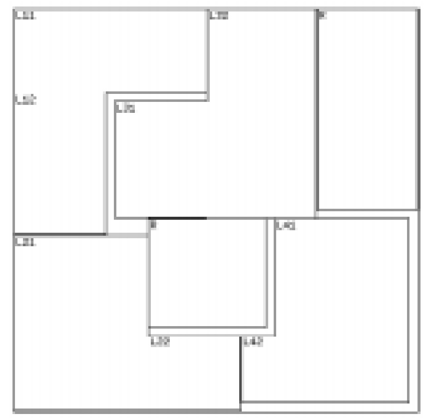
[8] H. Shin, A. L. Sangiovanni-Vincentelli, C. H. Sequin, “‘Zone-Refining’ Techniques for IC Layout Compaction”, *IEEE Trans. Computer-Aided Design*, vol. 9, Feb. 1990.

[9] D.F. Wong, C.L. Liu, “Floorplan Design for Rectangular and L-shaped Modules”, *Proc. IEEE International Conf. on Computer-Aided Design*, pp. 520-523, 1987.

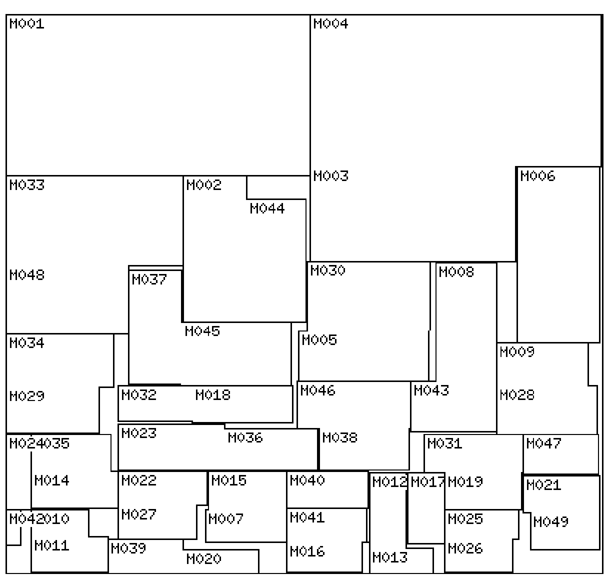
[10] B. T. Preas, W. M. vanCleemput, “Placement Algorithms for Arbitrarily Shaped Blocks”, *Proc. ACM/IEEE Design Automation Conf.*, pp. 474-480, 1979.

Table: The experimental results

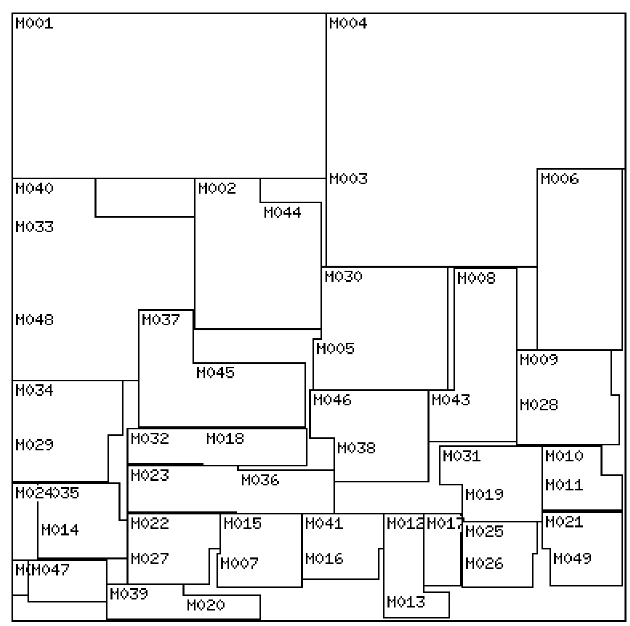
test case	# T-shaped	# L-shaped	# rectangular	area	dead space(%)
artificial	--	4	2	2304 (48 x 48)	5.16
ami49	--	21	7	37,391,508 (6314 x 5922)	5.20
ami49	1	20	6	39,424,616 (6314 x 6244)	10.09



(a) artificial test case(5.16%)



(b) modified *ami49*(5.20%)



(c) modified *ami49*(10.09%)

Fig. 14. Experimental results