Iterative Placement Improvement by Network Flow Methods

Konrad Doll, Frank M. Johannes, and Kurt J. Antreich, *Fellow, IEEE*

***Abstract*—We describe an efficient iterative improvement procedure for row-based cell placement with special emphasis on the objective function used to model net lengths. Two new net models are introduced and we prove theoretically that the net models are accurate approximations of the widely used half perimeter of a rectangle enclosing all pins of a net. In addition, unlike the half perimeter model, our net models allow us to compute costs for assigning cells to locations independently for all cells to be placed simultaneously. This offers our algorithm an important advantage compared to other iterative improvement techniques: many cells can be placed simultaneously by formulating placement as a network flow problem. This makes our algorithm more independent from a processing sequence than standard iterative improvement techniques. Finally, we compare our method to some existing algorithms including TimberWolfSC 5.4. We ran all of the algorithms on the SIGDA Benchmark Suite. We found that our method produced solutions with up to 23% less layout area while using an order of magnitude less running time compared to TimberWolfSC 5.4.**

I. Introduction

THERE ARE two challenging demands in automated layout synthesis of application specific integrated circuits (ASIC's). Firstly, for a *high-quality* layout the circuit's performance should be maximized and the chip area should be minimized. Secondly, layout design tools must be able to treat circuits with *complexities* of tens of thousands of cells. These two goals will become even more important with the expected increased usage of high-level synthesis tools. Either goal can only be approached at additional computing cost. For circuits of moderate complexities, current methods can satisfy both demands. But, for very large high-performance circuits, either excessive computation times have to be accepted or layout quality must be compromised.

To support the layout design of very large circuits, we have focused our research on placement algorithms, since it is during placement that the most crucial design decisions must be made. To both ease the placement and the routing tasks and allow the use of existing cell libraries, we adopt the popular row-oriented layout style with cells of equal heights and differing widths. This style is widely used for standard-cell circuits and for conventional gate arrays. It has also been successfully applied to the sea-of-gates layout style. For solving the placement problem, algorithms using constructive and iterative improvement techniques have been proposed [1], [2].

Constructive algorithms fall into two major classes, the *partitioning*-based algorithms [3], [4] and the *analytical* algorithms [5]–[7]. Recently, high-quality solutions were obtained with algorithms combining both strategies [8]–[11]. Constructive algorithms are usually fast and produce good results because of their global view of the problem. However, they are generally restricted in the choice of objectives and often do not yield the global optimum of the placement problem.

Iterative placement improvement algorithms aim at improving existing solutions, especially placements obtained with constructive algorithms. Typically, in one iterative step they select a small and local subproblem to be solved by exact or heuristic methods. These algorithms also divide into two classes depending upon whether they apply random or deterministic techniques.

Iterative improvement methods based on randomized algorithms never reject better solutions, but they also accept intermediate placements of inferior quality with low probabilities. Thus, they have the ability to escape local optima and to approach the global optimum arbitrarily close if sufficient computation time is provided. Since this is not always practicable, particularly for large circuits, layout quality is compromised. There are two basic randomized methods—Simulated Evolution [12], [13] and Simulated Annealing (SA) [14]–[16].

For algorithms applying the SA principle, it has been proved that they will provide a solution arbitrarily close to the global optimum if enough time is given. The semi-custom placement and routing package TimberWolfSC [16] is the dominant application that combines elaborate heuristics with the SA principle.

Deterministic placement improvement methods [17] usually offer the designer the choice in placement objectives, but they can get trapped in a local optimum. Since our research is particularly directed towards very large circuits, we concentrate on deterministic methods, which in general are much less computationally expensive than randomized methods.

The application of the exact and computationally efficient method of *linear assignment* has been proposed for solving placement problems with iterative techniques. It has been used previously to translate a global placement containing overlapping cells into an overlap-free final placement by minimizing the distance cells are moved away from their overlapping positions [18]. Linear assignment algorithms have also been proposed for solving the special placement problem, where all the cells have the same size and the possible

Manuscript received February 18, 1993; revised March 31, 1994. This paper was recommended by Editor Malgorzata Marek-Sadowska.

The authors are with the Department of Electrical Engineering, Institute of Electronic Design Automation, Technical University of Munich, 80290 Munich, Germany.

IEEE Log Number 9402559.

```
algorithm DOMINO
  initial_placement;
  while (improvement)
    generate_improved_placement;
  end
  adjust_row_lengths;
  swap_cells;
end
```

Fig. 1. Outline of the algorithm DOMINO.

locations are specified. In [19] cells of the same size are assigned to blocks in order to maximize the number of connections within the blocks. If the costs of assigning cells to locations are independent of each other, the placement task can be performed by iteratively selecting sets of unconnected cells and assigning them to locations [20], [21]. However, previous research experience with the unconnected sets technique indicates limited success [1], [17], [22] because components in highly interconnected groupings (a common occurrence in modern circuits) are not moved with respect to each other and nets with large numbers of pins tend to limit component movement.

Another linear assignment method [22] deliberately chooses a less sophisticated placement measure to avoid these difficulties. The number of adjacencies of connected cells is maximized by using a two-valued distance measure with the value 1 for adjacent connected cells and the value 0 for nonadjacent connected cells. In other methods [12], [23] nets connecting cells to be placed simultaneously are disregarded. Thus, if many cells are placed simultaneously, many nets are neglected and the assignment costs do not accurately reflect the total net length.

In this paper we show that the linear assignment method combined with an appropriate net model can overcome the above difficulties and can be applied successfully to determine high-quality placements for the row-based layout style. In addition, it can be easily modified for layout styles without a row structure.

Our method starts by applying the GORDIANL procedure [11]. The result is an initial placement with overlapping cells, that reflects the global structure of the circuit with high accuracy.

To improve the quality of this placement, we apply an iterative placement procedure called DOMINO [24]–[26]. To allow for cells of different sizes, the placement problem is generalized from the linear assignment problem to the *transportation problem*, which we solve by a network flow algorithm. Note that, unlike previous methods, we do not adopt the unconnected sets technique or neglect the nets connecting cells. Instead, we compute an improved placement for sets of cells, which may be connected and which are positioned near each other in the existing placement. To determine the transportation costs of cells to locations, we propose two new net models that are similar to the half perimeter of the minimum rectangle enclosing the pins of each net. Both models accurately reflect the lengths of nets connecting cells to be placed simultaneously.

Our paper is organized as follows. The next section contains a global outline of our iterative improvement procedure DOMINO. Section III describes the new net models used and shows the accuracy of these models in comparison to the half perimeter. In Section IV, results of benchmark examples are presented. Based on the final chip area obtained after final routing, comparisons with the simulated annealing method TimberWolfSC 5.4 [16], the quadrisection method [4] applied in VPNR, and GORDIANL are presented showing excellent performance of our placement tool.

II. Outline of the Procedure

The input to DOMINO consists of a net list, a cell library, and a description of the geometry of the chip. The net list is described by the sets $\mathcal{C}$ and $\mathcal{N}$ of the cells and the nets, respectively. All cells connected by net ν are in the set $\mathcal{C}_\nu$. All nets connected to a cell γ are in the set $\mathcal{N}_\gamma$. The placement algorithm DOMINO is shown in Fig. 1. It is composed of four main steps to be discussed in the following subsections.

A. Initial Placement

In the first step we determine an initial placement using the GORDIANL procedure [10], [11]. GORDIANL has been developed to place circuits with tens of thousands of cells. It is based on alternating global optimization and partitioning steps, thereby treating all cells simultaneously during all steps of partitioning. Wire length is minimized during global placement by solving a quadratic programming problem that has been proved to be efficiently solvable for examples with as many as 100,000 cells. The resulting placement will in general contain overlapping cells. It is often called *point*, *global* or *relative* placement, since the cells are treated as points and the placement reflects the optimal cell adjacencies in a global view. We use the point coordinates of the cells as an initial placement for our iterative improvement process.

B. Generation of an Improved Placement

The iterative process produces a sequence of intermediate placements. In each iterative step, an improved placement is generated from a current placement. After each generation, the placement is free of interspersed spaces and overlapping cells. The process terminates when after several generations no significant improvement is achieved.

Each generation of an improved placement is performed by solving a set of similar local subproblems. For that purpose the layout area is covered by an array of overlapping regions (Fig. 2). To each region ϱ we assign all cells currently placed inside that region. In the special case of the first generation, when the current placement is the point placement with possibly overlapping cells, we assign the cells to regions by recursively bipartitioning the set of cells with alternating horizontal and vertical cuts. Thus, even if the initial placement is a clustered placement, the cells are spread out over all regions. A subproblem consists of rearranging the cells currently placed inside a region. Since the cells have different widths, their rearrangement may produce overlaps and unused spaces. To construct a legal placement, we use

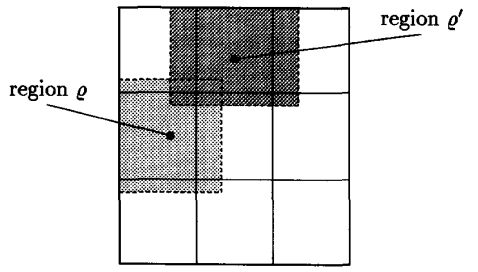


Fig. 2. Dividing the layout area into overlapping regions.

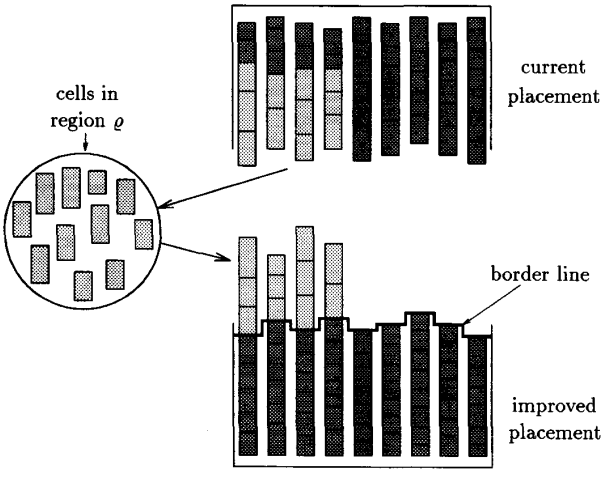


Fig. 3. Generation of an improved placement.

a simple but effective strategy—our constructed placements grow like crystals. A similar strategy has also been applied to compaction with success [27]. To preserve our analogy of growing the placement like a crystal, throughout the remainder of this paper we use a rotated view of the circuit. Thus, we take a column-oriented approach and have vertical rather than horizontal channels.

Suppose we are generating an improved placement from a current placement as shown in Fig. 3 by traversing through the regions from bottom to top. For the lower regions we have already produced an improved overlap-free and compact placement. The dark jagged line immediately above these lower regions is called the *border line*. The rearrangement process takes the cells in the next region ϱ and assigns them so that they abut the border line and are overlap free. The process is repeated on the remaining regions that are adjacent to the border line. Once the cells in all regions that are adjacent to the border line have been placed, the border line is moved to be immediately above the just-placed cells. The regions adjacent to the new border line are similarly processed. This process continues until the cells in all regions have been rearranged. Once all the regions have been processed, this *generation* of rearrangement is finished. In successive generations, different orderings of the regions are used. As the regions overlap each other, cells can move from one region to another during the rearrangement process.

```
algorithm generation
  for all regions ϱ
    get_cells;
    provide_locations;
    solve_transportation_problem;
    move_cells;
  end
end
```

Fig. 4. Outline of the algorithm *generation*.

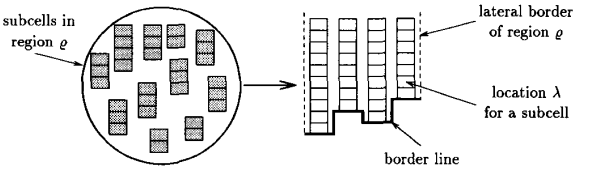


Fig. 5. Providing locations.

The pseudocode of the generation algorithm is shown in Fig. 4. In each region ϱ the cells are assigned to new positions by formulating a transportation problem, where units of cell area (subcells) are transported to locations according to a cost function approximating wire length. The four major operations of the algorithm are described below.

In each region we first determine the set of cells $\mathcal{C}_\varrho$ inside region ϱ with *get_cells*. To account for the different cell heights, we divide each cell $\mu \in \mathcal{C}_\varrho$ into s_μ subcells. Thus, the area of a cell is modeled by a suitable number of subcell area units. Typically, we choose the subcell size equal to the greatest common divisor of the cell sizes.

Next, we provide locations for all subcells above the border line between the vertical borders of region ϱ. Columns are filled with locations to form a straight line on top as shown in Fig. 5. The set of the locations is denoted by $\mathcal{L}_\varrho$.

We then simultaneously transport the subcells to locations in an overlap-free manner that minimizes the transportation cost. The transportation problem can be transformed into a minimal-cost maximum flow problem [2] on a network as shown in Fig. 6. This network consists of a source node S supplying subcells, a set of cell nodes μ, a set of location nodes λ, and a destination node D. The capacities of arcs between node S and cell nodes are s_μ such that cell μ can supply at most s_μ subcells. Since each location can hold at most one subcell, all capacities of arcs leading from location nodes to node D are set to one. The cost of assigning a subcell of cell μ to location λ is $c_{\mu\lambda}$. A detailed description and a theoretical analysis of the net model used to determine the cost $c_{\mu\lambda}$ follows in Section III. By using the flow augmentation method [28], [2] the procedure *solve_transportation_problem* can efficiently assign subcells to locations at minimum total transportation cost.

After solving the transportation problem we have assigned all subcells to locations. Since the same transportation cost is associated with all subcells of a cell and all subcells of a cell are pulled towards the cheapest location by the transportation algorithm, all subcells of a cell tend to lie side by side. This intuition was confirmed by our experimental results. Therefore,

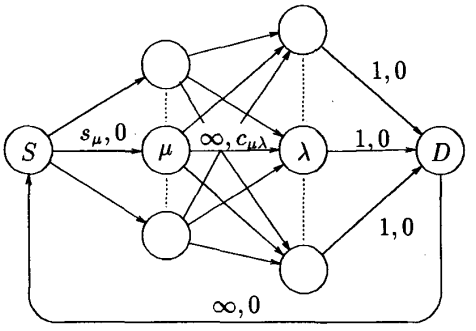


Fig. 6. Transportation network (arcs are labeled with capacity, cost).

in the procedure *move_cells* we place a cell in the column holding most of its subcells. In case of a tie, the cell is moved to the column containing the subcell, which causes the minimum transportation cost. The center of gravity of the subcells determines the y-coordinate of a cell. We stack the cells according to these y-coordinates, thus preventing overlap and unused space.

C. Adjustment of Column Lengths

After the generation process, the column lengths generally differ no more than 5% from the average column length. The goal of column lengths adjustment is to make sure that all columns will have equal lengths after final routing. This computation includes an estimation of the feedthroughs for the various columns. To get columns of equal lengths as far as possible, few cells are exchanged between neighboring columns. This usually results in a $1 \ldots 2\%$ increase of the estimated wire length.

D. Swapping of Cells

In the final step of the iterative improvement process, we only perform intra-column swaps limiting region width to one column. Since the rearrangement of adjacent cells in a column does not produce any overlaps and unused spaces, a legal placement is generated after each call of *move_cells*. To construct a highly optimized placement, only rearrangements are accepted that result in a decreased estimated wire length.

III. New Net Models

At the beginning of this section, we show how the transportation costs are computed. For this purpose two new net models—net model I and net model II—are introduced. Finally, these two net models are analyzed theoretically.

For the computation of the transportation costs and the description of the new net models, we need some definitions. Let (x_γ, y_γ) denote the center coordinates of cell γ. To simplify the following discussion, we assume that all nets are connected to the center of the cell. Of course, we consider the real pin positions in DOMINO. We define the disjoint sets $\mathcal{O}_\nu = \mathcal{C}_\nu \backslash \mathcal{C}_\varrho$ and $\mathcal{I}_\nu = \mathcal{C}_\nu \cap \mathcal{C}_\varrho$ containing all cells connected to net ν outside and inside region ϱ, respectively. The cells in $\mathcal{O}_\nu$ and $\mathcal{I}_\nu$ are denoted by $\mathcal{O}_\nu$-cells and $\mathcal{I}_\nu$-cells, respectively. Furthermore, let $\mathbf{x} = [\ldots, x_\gamma, \ldots]^T$ and $\mathbf{y} = [\ldots, y_\gamma, \ldots]^T$ denote the vectors with the unknown coordinates of all cells $\gamma \in \mathcal{C}_\varrho$. The coordinates of all cells outside region ϱ are known and do not change when processing region ϱ. The vectors $\mathbf{x}_\nu = [\ldots, x_\gamma, \ldots]^T$ and $\mathbf{y}_\nu = [\ldots, y_\gamma, \ldots]^T$ contain the unknown coordinates of the $\mathcal{I}_\nu$-cells. Let $\underline{\mathbf{x}}_\nu = [\ldots, \underline{x}_\gamma, \ldots]^T$ and $\underline{\mathbf{y}}_\nu = [\ldots, \underline{y}_\gamma, \ldots]^T$ denote the vectors with the known coordinates $(\underline{x}_\gamma, \underline{y}_\gamma)$ of the $\mathcal{I}_\nu$-cells from the current placement. The half perimeter L_ν of the minimum rectangle enclosing all pins of net ν is

$$L_\nu = \max_{\gamma \in \mathcal{C}_\nu}\{x_\gamma\} - \min_{\gamma \in \mathcal{C}_\nu}\{x_\gamma\} + \max_{\gamma \in \mathcal{C}_\nu}\{y_\gamma\} - \min_{\gamma \in \mathcal{C}_\nu}\{y_\gamma\}. \quad (1)$$

A. Transportation Costs

Before the transportation problem is solved the cost $c_{\mu\lambda}$ of assigning a subcell of cell $\mu \in \mathcal{C}_\varrho$ to location $\lambda \in \mathcal{L}_\varrho$ with the coordinates (x_λ, y_λ) is calculated by

$$c_{\mu\lambda} = \sum_{\nu \in \mathcal{N}_\mu} \Gamma_{\mu\nu}(x_\mu, y_\mu) \text{ with } x_\mu = x_\lambda, y_\mu = y_\lambda, \quad (2)$$

where $\Gamma_{\mu\nu}(x_\mu, y_\mu)$ is the wire length estimate of net ν.

The cost $c_{\mu\lambda}$ of the cell μ has to be independent from the positions of the cells $\gamma \in \mathcal{C}_\varrho \backslash \{\mu\}$ (all cells in region ϱ except cell μ), because their positions are unknown when calculating the cost $c_{\mu\lambda}$. This means that $\Gamma_{\mu\nu}$ has to be independent from the positions of the cells $\gamma \in \mathcal{I}_\nu \backslash \{\mu\}$ (all cells in region ϱ connected to net ν except cell μ), i.e.

$$\Gamma_{\mu\nu}(x_\mu, y_\mu) \neq \Gamma_{\mu\nu}(x_\gamma, y_\gamma), \quad \gamma \in \mathcal{I}_\nu \backslash \{\mu\}. \quad (3)$$

Since the half perimeter depends on the positions of all $\mathcal{I}_\nu$-cells ($L_\nu = L_\nu(\mathbf{x}_\nu, \mathbf{y}_\nu)$), the half perimeter $L_\nu(\mathbf{x}_\nu, \mathbf{y}_\nu)$ is generally not suitable for the computation of $\Gamma_{\mu\nu}(x_\mu, y_\mu)$. Therefore, we approximate the half perimeter with two new net models that satisfy (3). Then, the *optimal* positions of the subcells with respect to these net models can easily be determined by a transportation algorithm. The accuracy of both net models with respect to the half perimeter will be discussed in Section III.B.

For the following computation of $\Gamma_{\mu\nu}(x_\mu, y_\mu)$, three net types will be considered depending on the number of $\mathcal{I}_\nu$-cells:

- *Type A:* $|\mathcal{I}_\nu| = 1$
- *Type B:* $1 < |\mathcal{I}_\nu| < |\mathcal{C}_\nu|$
- *Type C:* $|\mathcal{I}_\nu| = |\mathcal{C}_\nu|$

Introduction of Net Model I:

Type A: This type applies to $|\mathcal{I}_\nu| = 1$, i.e. net ν connects only one $\mathcal{I}_\nu$-cell μ with some $\mathcal{O}_\nu$-cells. Then, the wire length estimate $\Gamma_{\mu\nu}(x_\mu, y_\mu)$ does not depend on the positions of other cells in region ϱ and $\Gamma_{\mu\nu}(x_\mu, y_\mu)$ can be calculated exactly:

$$\Gamma_{\mu\nu}(x_\mu, y_\mu) = L_\nu(x_\mu, y_\mu) \quad (4)$$

Type B: This type applies to $1 < |\mathcal{I}_\nu| < |\mathcal{C}_\nu|$ i.e. net ν connects more than one $\mathcal{I}_\nu$-cell with one or more $\mathcal{O}_\nu$-cells. This type is illustrated in Fig. 7, where net ν connects the $\mathcal{I}_\nu$-cell μ with the $\mathcal{I}_\nu$-cells α and β, and the $\mathcal{O}_\nu$-cells ϵ and ζ. Then $\mathcal{I}_\nu = \{\mu, \alpha, \beta\}$ and $\mathcal{O}_\nu = \{\epsilon, \zeta\}$.

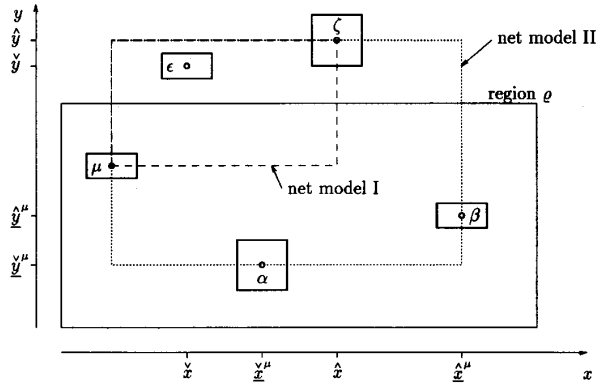


Fig. 7. Net ν connecting the cells $\mu, \alpha, \beta, \epsilon$, and ζ.

To make the cost values independent, the first net model I neglects the interconnection between the $\mathcal{I}_\nu$-cells and calculates the cost values as if each cell was connected only to the $\mathcal{O}_\nu$-cells. With the lower and upper bounds for the coordinates of the $\mathcal{O}_\nu$-cells

$$\begin{aligned} \check{x} &= \min_{\gamma \in \mathcal{O}_\nu} \{x_\gamma\}, \quad \hat{x} = \max_{\gamma \in \mathcal{O}_\nu} \{x_\gamma\}, \\ \check{y} &= \min_{\gamma \in \mathcal{O}_\nu} \{y_\gamma\}, \quad \hat{y} = \max_{\gamma \in \mathcal{O}_\nu} \{y_\gamma\}, \end{aligned} \tag{5}$$

the contribution of net ν to the cost $c_{\mu\lambda}$ is

$$\begin{aligned} \Gamma_{\mu\nu}(x_\mu, y_\mu) = &\max(\hat{x}, x_\mu) - \min(\check{x}, x_\mu) \\ &+ \max(\hat{y}, y_\mu) - \min(\check{y}, y_\mu). \end{aligned} \tag{6}$$

For the net in Fig. 7, the lower and upper bounds $\check{x} = \min(x_\epsilon, x_\zeta)$, $\hat{x} = \max(x_\epsilon, x_\zeta)$, $\check{y} = \min(y_\epsilon, y_\zeta)$, $\hat{y} = \max(y_\epsilon, y_\zeta)$, are marked on the x- and y-axes. The enclosing rectangle calculated by (6) is illustrated by dashed lines.

Type C: This type applies to $|\mathcal{I}_\nu| = |\mathcal{C}_\nu|$, i.e. net ν connects only $\mathcal{I}_\nu$-cells. To prevent net ν from being neglected, which might increase net length, we introduce a virtual cell ϕ. All $\mathcal{I}_\nu$ -cells are temporarily connected to cell ϕ, which is positioned in the center of gravity of the $\mathcal{I}_\nu$-cells. The center of gravity is calculated with respect to the current placement:

$$x_\phi = \frac{1}{|\mathcal{I}_\nu|} \sum_{\mu \in \mathcal{I}_\nu} \underline{x}_\mu, \quad y_\phi = \frac{1}{|\mathcal{I}_\nu|} \sum_{\mu \in \mathcal{I}_\nu} \underline{y}_\mu \tag{7}$$

We obtain the contribution of net ν to the cost $c_{\mu\lambda}$ from (5) and (6) with $\mathcal{O}_\nu = \{\phi\}$:

$$\begin{aligned} \Gamma_{\mu\nu}(x_\mu, y_\mu) = &\max(x_\phi, x_\mu) - \min(x_\phi, x_\mu) \\ &+ \max(y_\phi, y_\mu) - \min(y_\phi, y_\mu) \end{aligned} \tag{8}$$

In Fig. 8 we give an example for this type, i.e. a net ν connecting the cell μ with the cells α and β. The enclosing rectangle calculated by (8) is shown by dashed lines.

To summarize all three net types, we can compute $\Gamma_{\mu\nu}(x_\mu, y_\mu)$ after a suitable choice of $\mathcal{O}_\nu$ for all nets with

$$\begin{aligned} \Gamma_{\mu\nu}(x_\mu, y_\mu) = &\max(\hat{x}, x_\mu) - \min(\check{x}, x_\mu) \\ &+ \max(\hat{y}, y_\mu) - \min(\check{y}, y_\mu). \end{aligned} \tag{9}$$

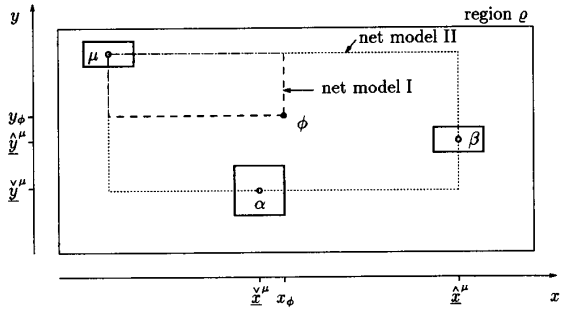


Fig. 8. Net ν connecting the cells μ, α, and β.

Introduction of Net Model II:

Type A ($|\mathcal{I}_\nu| = 1$): As for net model I, the contribution $\Gamma_{\mu\nu}(x_\mu, y_\mu)$ can be calculated exactly:

$$\Gamma_{\mu\nu}(x_\mu, y_\mu) = L_\nu(x_\mu, y_\mu) \tag{10}$$

Type B ($1 < |\mathcal{I}_\nu| < |\mathcal{C}_\nu|$): In contrary to net model I, which neglects the interconnection between the $\mathcal{I}_\nu$-cells (i.e. cells μ, α, and β in Fig. 7), net model II estimates the unknown coordinates of the cells $\gamma \in \mathcal{I}_\nu \backslash \{\mu\}$ at their coordinates $(\underline{x}_\gamma, \underline{y}_\gamma)$ from the current placement. With the lower and upper bounds for the estimated coordinates of the cells $\gamma \in \mathcal{I}_\nu \backslash \{\mu\}$

$$\begin{aligned} \underline{\check{x}}^\mu &= \min_{\gamma \in \mathcal{I}_\nu \backslash \{\mu\}} \{\underline{x}_\gamma\}, \quad \underline{\hat{x}}^\mu = \max_{\gamma \in \mathcal{I}_\nu \backslash \{\mu\}} \{\underline{x}_\gamma\}, \\ \underline{\check{y}}^\mu &= \min_{\gamma \in \mathcal{I}_\nu \backslash \{\mu\}} \{\underline{y}_\gamma\}, \quad \underline{\hat{y}}^\mu = \max_{\gamma \in \mathcal{I}_\nu \backslash \{\mu\}} \{\underline{y}_\gamma\}, \end{aligned} \tag{11}$$

and the lower and upper bounds for the coordinates of the $\mathcal{O}_\nu$-cells (5) the contribution of net ν to the cost $c_{\mu\lambda}$ is

$$\begin{aligned} \Gamma_{\mu\nu}(x_\mu, y_\mu) = &\max(\hat{x}, \underline{\hat{x}}^\mu, x_\mu) - \min(\check{x}, \underline{\check{x}}^\mu, x_\mu) \\ &+ \max(\hat{y}, \underline{\hat{y}}^\mu, y_\mu) - \min(\check{y}, \underline{\check{y}}^\mu, y_\mu). \end{aligned} \tag{12}$$

In Fig. 7 the lower and upper bounds $\underline{\check{x}}^\mu = \min(\underline{x}_\alpha, \underline{x}_\beta)$, $\underline{\hat{x}}^\mu = \max(\underline{x}_\alpha, \underline{x}_\beta)$, $\underline{\check{y}}^\mu = \min(\underline{y}_\alpha, \underline{y}_\beta)$, and $\underline{\hat{y}}^\mu = \max(\underline{y}_\alpha, \underline{y}_\beta)$ are also marked. The enclosing rectangle defined by (12) is illustrated by dotted lines.

Type C ($|\mathcal{I}_\nu| = |\mathcal{C}_\nu|$): Again we introduce the virtual cell ϕ, set $\mathcal{O}_\nu = \{\phi\}$, and estimate the unknown coordinates of the cells $\gamma \in \mathcal{I}_\nu \backslash \{\mu\}$ at their coordinates $(\underline{x}_\gamma, \underline{y}_\gamma)$ from the current placement. Then the contribution of net ν to the cost $c_{\mu\lambda}$ is

$$\begin{aligned} \Gamma_{\mu\nu}(x_\mu, y_\mu) = &\max(x_\phi, \underline{\hat{x}}^\mu, x_\mu) - \min(x_\phi, \underline{\check{x}}^\mu, x_\mu) \\ &+ \max(y_\phi, \underline{\hat{y}}^\mu, y_\mu) - \min(y_\phi, \underline{\check{y}}^\mu, y_\mu). \end{aligned} \tag{13}$$

The enclosing rectangle calculated by (13) is illustrated by dotted lines in Fig. 8. To summarize the three types of net model II, we can compute $\Gamma_{\mu\nu}(x_\mu, y_\mu)$ after a suitable choice of $\mathcal{O}_\nu$ for all nets with

$$\begin{aligned} \Gamma_{\mu\nu}(x_\mu, y_\mu) = &\max(\hat{x}, \underline{\hat{x}}^\mu, x_\mu) - \min(\check{x}, \underline{\check{x}}^\mu, x_\mu) \\ &+ \max(\hat{y}, \underline{\hat{y}}^\mu, y_\mu) - \min(\check{y}, \underline{\check{y}}^\mu, y_\mu). \end{aligned} \tag{14}$$

Determination of the Cost Functions From the New Net Models: For an in-depth analysis of the two new net models in the following Section III.B, we need the contribution of a single net to the *objective function* $\Phi(\mathbf{x},\mathbf{y})$ of the transportation problem. The objective function is (see (2))

$$\Phi(\mathbf{x},\mathbf{y}) = \sum_{\mu\in\mathcal{C}_\varrho} c_{\mu\lambda} = \sum_{\mu\in\mathcal{C}_\varrho}\sum_{\nu\in\mathcal{N}_\mu} \Gamma_{\mu\nu}(x_\mu, y_\mu). \tag{15}$$

With the set $\mathcal{N}_\varrho = \bigcup_{\mu\in\mathcal{C}_\varrho}\mathcal{N}_\mu$ of nets connected to cells in region ϱ (15) can be written as

$$\Phi(\mathbf{x},\mathbf{y}) = \sum_{\nu\in\mathcal{N}_\varrho}\sum_{\mu\in\mathcal{I}_\nu} \Gamma_{\mu\nu}(x_\mu, y_\mu). \tag{16}$$

In (16) we summarize the terms $\sum_{\mu\in\mathcal{I}_\nu}\Gamma_{\mu\nu}(x_\mu,y_\mu)$ over all nets $\nu\in\mathcal{N}_\varrho$. Each such term can be viewed as the *cost function* ℓ_ν of net ν:

$$\ell_\nu(\mathbf{x}_\nu,\mathbf{y}_\nu) = \sum_{\mu\in\mathcal{I}_\nu} \Gamma_{\mu\nu}(x_\mu, y_\mu). \tag{17}$$

For the nets in Figs. 7 and 8 we get the cost function $\ell_\nu(\mathbf{x}_\nu,\mathbf{y}_\nu)$ by summing $\Gamma_{\mu\nu}(x_\mu,y_\mu)$, $\Gamma_{\alpha\nu}(x_\alpha,y_\alpha)$, $\Gamma_{\beta\nu}(x_\beta,y_\beta)$.

From (9) and (17) we obtain the cost function $\ell_\nu(\mathbf{x}_\nu,\mathbf{y}_\nu)$ of net model I

$$\begin{aligned}\ell_\nu(\mathbf{x}_\nu,\mathbf{y}_\nu) = \sum_{\mu\in\mathcal{I}_\nu} \{&\max(\hat{x},x_\mu) - \min(\check{x},x_\mu)\\ &+\max(\hat{y},y_\mu) - \min(\check{y},y_\mu)\}.\end{aligned} \tag{18}$$

Similarly we get the cost function $\ell_\nu(\mathbf{x}_\nu,\mathbf{y}_\nu)$ of net model II from (14) and (17)

$$\begin{aligned}\ell_\nu(\mathbf{x}_\nu,\mathbf{y}_\nu) = \sum_{\mu\in\mathcal{I}_\nu} \Big\{&\max(\hat{x},\underline{\hat{x}}^\mu,x_\mu) - \min(\check{x},\underline{\check{x}}^\mu,x_\mu)\\ &+\max(\hat{y},\underline{\hat{y}}^\mu,y_\mu) - \min(\check{y},\underline{\check{y}}^\mu,y_\mu)\Big\}.\end{aligned} \tag{19}$$

B. Analysis of the New Net Models

In this subsection the cost functions $\ell_\nu(\mathbf{x}_\nu,\mathbf{y}_\nu)$ for the net models I and II are compared with the half perimeter $L_\nu(\mathbf{x}_\nu,\mathbf{y}_\nu)$ of a net ν. Since the same arguments hold for x- and y-coordinates, we consider only x-coordinates. The x-part of the half perimeter is (see (1))

$$L_\nu(\mathbf{x}_\nu) = \max_{\gamma\in\mathcal{C}_\nu}\{x_\gamma\} - \min_{\gamma\in\mathcal{C}_\nu}\{x_\gamma\}. \tag{20}$$

From (18) and (19) we obtain the x-part of the cost function for net model I

$$\ell_\nu(\mathbf{x}_\nu) = \sum_{\mu\in\mathcal{I}_\nu}\{\max(\hat{x},x_\mu) - \min(\check{x},x_\mu)\} \tag{21}$$

and for net model II

$$\ell_\nu(\mathbf{x}_\nu) = \sum_{\mu\in\mathcal{I}_\nu}\{\max(\hat{x},\underline{\hat{x}}^\mu,x_\mu) - \min(\check{x},\underline{\check{x}}^\mu,x_\mu)\}. \tag{22}$$

The half perimeter $L_\nu(\mathbf{x}_\nu)$ as well as the cost function $\ell_\nu(\mathbf{x}_\nu)$ depends on the unknown coordinates of the $\mathcal{I}_\nu$-cells. At each point $\mathbf{x}_\nu\in\mathbb{R}^{|\mathcal{I}_\nu|}$, that means for all possible positions of the $\mathcal{I}_\nu$-cells, we can compute the function values $L_\nu(\mathbf{x}_\nu)$ and $\ell_\nu(\mathbf{x}_\nu)$. In addition to the function values, we can evaluate the gradients of $L_\nu(\mathbf{x}_\nu)$ and $\ell_\nu(\mathbf{x}_\nu)$, $\mathbf{g}_L(\mathbf{x}_\nu)$ and $\mathbf{g}_\ell(\mathbf{x}_\nu)$, at each point $\mathbf{x}_\nu\in\mathbb{R}^{|\mathcal{I}_\nu|}$, where $L_\nu(\mathbf{x}_\nu)$ and $\ell_\nu(\mathbf{x}_\nu)$ are differentiable. In the following sections these functions and gradients will be theoretically discussed. The important results are:

1) Net model I has the property that the gradient of the cost function $\mathbf{g}_\ell(\mathbf{x}_\nu)$ has no component pointing opposing the gradient of the half perimeter $\mathbf{g}_L(\mathbf{x}_\nu)$. Thus, when minimizing the cost function $\ell_\nu(\mathbf{x}_\nu)$ by decreasing along the axis of the steepest descent, the half perimeter $L_\nu(\mathbf{x}_\nu)$ is never increased.
2) For the second net model II, the gradients of the cost function and the half perimeter are equal in the point of the current placement, i.e. $\mathbf{g}_L(\mathbf{x}_\nu = \underline{\mathbf{x}}_\nu) = \mathbf{g}_\ell(\mathbf{x}_\nu = \underline{\mathbf{x}}_\nu)$. Therefore, this net model is an accurate approximation of the half perimeter in the neighborhood of the current placement. A similar strategy is also used in nonlinear optimization. For example, in each iterative step of the *steepest descent method* [29], a nonlinear function is approximated by a linear function whose first derivative is equal to the gradient of the nonlinear function in the current point.

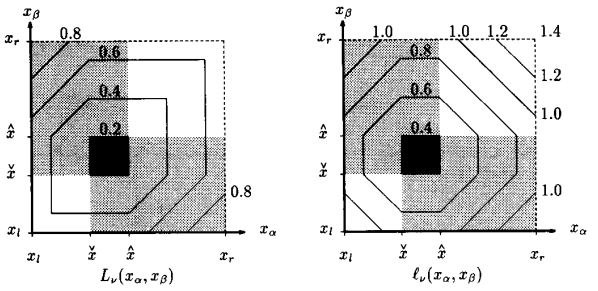


Fig. 9. Net model I: Level contours of L_ν and ℓ_ν for $|\mathcal{I}_\nu| = 2 \wedge |\mathcal{I}_\nu| < |\mathcal{C}_\nu|$.

Analysis of Net Model I: To illustrate the relation between $L_\nu(\mathbf{x}_\nu)$ and $\ell_\nu(\mathbf{x}_\nu)$, we will first discuss the case $|\mathcal{I}_\nu| = 2$ and then the general case with arbitrary $|\mathcal{I}_\nu|$. The cost value of a net with $|\mathcal{I}_\nu| = 1$ is calculated exactly (Type A).

Suppose a net connects two $\mathcal{I}_\nu$-cells α,β with some $\mathcal{O}_\nu$-cells (Type B). Equations (20) and (21) can be written as

$$L_\nu(x_\alpha,x_\beta) = \max(\hat{x},x_\alpha,x_\beta) - \min(\check{x},x_\alpha,x_\beta) \tag{23}$$

$$\begin{aligned}\ell_\nu(x_\alpha,x_\beta) = &\max(\hat{x},x_\alpha) - \min(\check{x},x_\alpha)\\ &+\max(\hat{x},x_\beta) - \min(\check{x},x_\beta).\end{aligned} \tag{24}$$

The level contours of both functions are illustrated in Fig. 9, where x_l and x_r are the coordinates of the left and right border of region ϱ, respectively. The function values labeling the level contours are scaled to $(x_r - x_l)$.

In the shaded regions of Fig. 9 the difference $L_\nu(x_\alpha,x_\beta) - \ell_\nu(x_\alpha,x_\beta)$ is constant. This means that the gradients of the functions $L_\nu(x_\alpha,x_\beta)$ and $\ell_\nu(x_\alpha,x_\beta)$ are equal, if one cell's x-coordinate is smaller than $\hat{x}$ and the other larger than $\check{x}$. Furthermore, the optimum points (marked black in Fig. 9) of the cost function $\ell_\nu(x_\alpha,x_\beta)$ are identical with the optimum

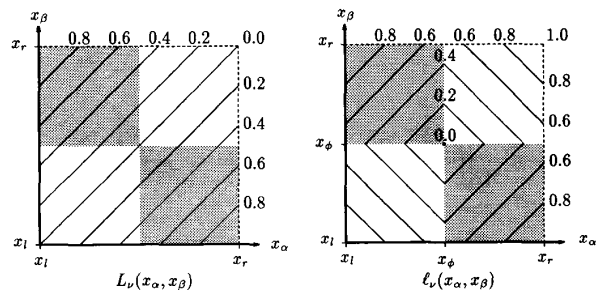


Fig. 10. Net model I: Level contours of L_ν and ℓ_ν for $|\mathcal{I}_\nu| = |\mathcal{C}_\nu|$.

points of $L_\nu(x_\alpha, x_\beta)$. The gradients of the two functions differ only outside the shaded regions, where the coordinates of cells α and β are both below $\check{x}$ or both above $\hat{x}$. The angle between the gradients of both functions is 45° outside the shaded regions. Thus, when minimizing $\ell_\nu(x_\alpha, x_\beta)$ by decreasing along the direction of the steepest descent of $\ell_\nu(x_\alpha, x_\beta)$ we also reduce $L_\nu(x_\alpha, x_\beta)$. If $x_l > \hat{x}$ or $x_r < \check{x}$, the optimum points of both functions are also identical and the angle between the gradients is 45°.

For nets connecting only two $\mathcal{I}_\nu$-cells (Type C) we get

$$L_\nu(x_\alpha, x_\beta) = \max(x_\alpha, x_\beta) - \min(x_\alpha, x_\beta) \tag{25}$$

$$\ell_\nu(x_\alpha, x_\beta) = \max(x_\phi, x_\alpha) - \min(x_\phi, x_\alpha) + \max(x_\phi, x_\beta) - \min(x_\phi, x_\beta). \tag{26}$$

from (20) and (21). The virtual cell ϕ always lies inside the region $x_l \le x_\phi \le x_r$. Fig. 10 shows the level contours of $L_\nu(x_\alpha, x_\beta)$ and $\ell_\nu(x_\alpha, x_\beta)$. $L_\nu(x_\alpha, x_\beta)$ and $\ell_\nu(x_\alpha, x_\beta)$ are equal in the shaded regions, i.e. the functions are equal, if one cell lies on the left-hand side and the other on the right-hand side of cell ϕ. Further, the optimum of $\ell_\nu(x_\alpha, x_\beta)$ is contained in the set of optimum points of $L_\nu(x_\alpha, x_\beta)$. For the points where $\ell_\nu(x_\alpha, x_\beta) \neq L_\nu(x_\alpha, x_\beta)$, the gradients of both functions are orthogonal to each other.

Now we will discuss the relation between the gradients of $L_\nu(\mathbf{x}_\nu)$ and $\ell_\nu(\mathbf{x}_\nu)$ in the general case.

Theorem 1: The scalar product of the gradients of $L_\nu(\mathbf{x}_\nu)$ and $\ell_\nu(\mathbf{x}_\nu)$, $\mathbf{g}_\ell(\mathbf{x}_\nu)$ and $\mathbf{g}_\ell(\mathbf{x}_\nu)$, is always greater or equal 0, i.e. $\mathbf{g}_L^T(\mathbf{x}_\nu) \cdot \mathbf{g}_\ell(\mathbf{x}_\nu) \ge 0$.

Proof: We will prove this for the Types A, B, C separately, by calculating first the two gradients and then the scalar product.

Type A ($|\mathcal{I}_\nu| = 1$): For nets with $|\mathcal{I}_\nu| = 1$ the cost function $\ell_\nu(\mathbf{x}_\nu)$ and the half perimeter $L_\nu(\mathbf{x}_\nu)$ are identical. Thus, the gradients are also identical: $\mathbf{g}_L(\mathbf{x}_\nu) = \mathbf{g}_\ell(\mathbf{x}_\nu)$. For the scalar product we get

$$\mathbf{g}_L^T(\mathbf{x}_\nu) \cdot \mathbf{g}_\ell(\mathbf{x}_\nu) = |\mathbf{g}_L(\mathbf{x}_\nu)|^2 = |\mathbf{g}_l(\mathbf{x}_\nu)|^2 \ge 0. \tag{27}$$

Type B ($1 < |\mathcal{I}_\nu| < |\mathcal{C}_\nu|$): Suppose we are given a net ν with $\mathcal{I}_\nu = \{\alpha, \ldots, \mu, \ldots, \beta\}$ and $|\mathcal{I}_\nu| < |\mathcal{C}_\nu|$. Without loss of generality, we consider the case $x_\alpha < \cdots < x_\mu < \cdots < x_\beta$. All other coordinate orderings can be reduced to this case. From (20) follows

$$L_\nu(\mathbf{x}_\nu) = \max\Big(\max_{\gamma \in \mathcal{I}_\nu}\{x_\gamma\}, \max_{\gamma \in \mathcal{O}_\nu}\{x_\gamma\}\Big) - \min\Big(\min_{\gamma \in \mathcal{I}_\nu}\{x_\gamma\}, \min_{\gamma \in \mathcal{O}_\nu}\{x_\gamma\}\Big). \tag{28}$$

Since $x_\alpha < \cdots < x_\mu < \cdots < x_\beta$ we obtain

$$\max_{\gamma \in \mathcal{I}_\nu}\{x_\gamma\} = x_\beta \quad \text{and} \quad \min_{\gamma \in \mathcal{I}_\nu}\{x_\gamma\} = x_\alpha. \tag{29}$$

With (5) and (29), (28) can be written as

$$L_\nu(\mathbf{x}_\nu) = L_\nu(x_\alpha, x_\beta) = \max(x_\beta, \hat{x}) - \min(x_\alpha, \check{x}). \tag{30}$$

Then $\mathbf{g}_L(\mathbf{x}_\nu) = \mathbf{g}_L(x_\alpha, x_\beta) = [\mathbf{g}_{L\alpha}(x_\alpha), 0, \ldots, 0, g_{L\beta}(x_\beta)]^T$ with

$$g_{L\alpha}(x_\alpha) = \begin{cases} -1 & \text{if } x_\alpha < \check{x} \\ 0 & \text{if } x_\alpha > \check{x} \end{cases}$$

$$\text{and } g_{L\beta}(x_\beta) = \begin{cases} 0 & \text{if } x_\beta < \hat{x} \\ 1 & \text{if } x_\beta > \hat{x} \end{cases}. \tag{31}$$

At the points $x_\alpha = \check{x}$ and $x_\beta = \hat{x}$ the gradient does not exist.

When calculating $\mathbf{g}_\ell(\mathbf{x}_\nu) = [g_{\ell\alpha}(x_\alpha), \ldots, g_{\ell\mu}(x_\mu), \ldots, g_{\ell\beta}(x_\beta)]^T$ from (21), we get

$$g_{\ell\mu}(x_\mu) = \begin{cases} -1 & \text{if } x_\mu < \check{x} \\ 0 & \text{if } \check{x} < x_\mu < \hat{x} \\ 1 & \text{if } \hat{x} < x_\mu \end{cases} \tag{32}$$

for all $\mu \in \mathcal{I}_\nu$. In the case $\check{x} < x_\alpha < x_\beta < \hat{x}$ this implies that $\mathbf{g}_L = \mathbf{g}_\ell = [0, 0, \ldots, 0, 0]^T$, and the scalar product $\mathbf{g}_L^T \cdot \mathbf{g}_\ell = 0$. In all other cases simple arithmetic shows that $\mathbf{g}_L^T \cdot \mathbf{g}_\ell(\mathbf{x}_\nu) = g_{L\alpha}(x_\alpha) g_{\ell\alpha}(x_\alpha) + g_{L\beta}(x_\beta) g_{\ell\beta}(x_\beta) > 0$ for all x_α and x_β.

Type C ($|\mathcal{I}_\nu| = |\mathcal{C}_\nu|$): For nets with $|\mathcal{I}_\nu| = |\mathcal{C}_\nu|$, $\mathbf{g}_L(\mathbf{x}_\nu) = \mathbf{g_L} = [g_{L\alpha}, \mathbf{0}, \ldots, \mathbf{0}, g_{L\beta}]^T$ with $g_{L\alpha} = -1$ and $g_{L\beta} = 1$ follows from (20). From (21) we get $\mathbf{g}_\ell(\mathbf{x}_\nu) = [g_{\ell\alpha}(\mathbf{x}_\alpha), \ldots, g_{\ell\mu}(\mathbf{x}_\mu), \ldots g_{\ell\beta}(\mathbf{x}_\beta)]^T$ with

$$g_{\ell\mu}(x_\mu) = \begin{cases} -1 & \text{if } x_\mu < x_\phi \\ 1 & \text{if } x_\mu > x_\phi \end{cases} \tag{33}$$

for all $\mu \in \mathcal{I}_\nu$. In the cases $x_\alpha < x_\beta < x_\phi$ (all cells lie on the left-hand side of cell ϕ) and $x_\phi < x_\alpha < x_\beta$ (all cells lie on the right-hand side of cell ϕ) the scalar product $\mathbf{g}_L^T \cdot \mathbf{g}_\ell(\mathbf{x}_\nu) = \mathbf{0}$, where the Euclidean norm of neither $\mathbf{g}_L$ nor $\mathbf{g}_\ell(\mathbf{x}_\nu)$ is 0. Thus, the gradients are orthogonal to each other.

In the case $x_\alpha < x_\phi < x_\beta$ (at least one cell lies on the left-hand side and one on the right-hand side of cell ϕ) the scalar product $\mathbf{g}_L^T \cdot \mathbf{g}_\ell(\mathbf{x}_\nu) = 2$, and the angle φ between the gradients depends on the Euclidean norm of $\mathbf{g}_\ell$ and $\mathbf{g}_L(\mathbf{x}_\nu)$. Since $g_{L\alpha} = -1$ and $g_{L\beta} = 1$, we obtain the Euclidean norm $|\mathbf{g}_L| = \sqrt{2}$. Because all components of $\mathbf{g}_\ell$ are $+1$ or -1, $|\mathbf{g}_\ell(\mathbf{x}_\nu)| = \sqrt{|\mathcal{I}_\nu|}$. The relation between the angle φ and the gradients is

$$\cos\varphi = \frac{\mathbf{g}_L^T \cdot \mathbf{g}_\ell(\mathbf{x}_\nu)}{|\mathbf{g}_L| \cdot |\mathbf{g}_\ell(\mathbf{x}_\nu)|} = \frac{2}{\sqrt{2} \cdot \sqrt{|\mathcal{I}_\nu|}} = \frac{\sqrt{2}}{\sqrt{|\mathcal{I}_\nu|}}. \tag{34}$$

Equation (34) shows that the angle φ lies between 0° for nets connecting two cells and 90° for nets connecting an infinite number of cells. □

Since $\mathbf{g}_L^T(\mathbf{x}_\nu) \cdot \mathbf{g}_\ell(\mathbf{x}_\nu) \ge 0$, the angle between the gradients does not exceed 90°. Therefore, we are able to decompose

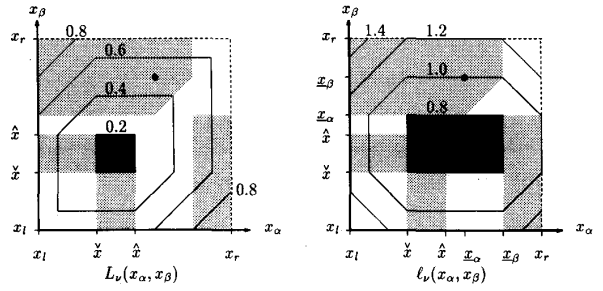


Fig. 11. Net model II: Level contours of L_ν and ℓ_ν for $|\mathcal{I}_\nu| = 2 \wedge |\mathcal{I}_\nu| < |\mathcal{C}_\nu|$.

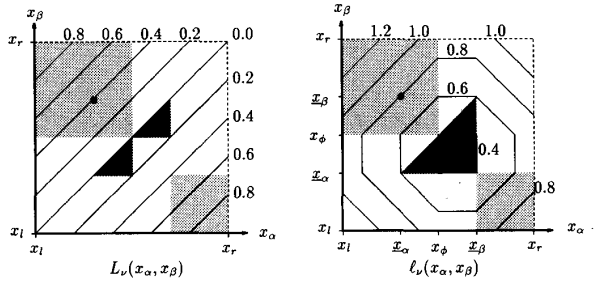


Fig. 12. Net model II: Level contours of L_ν and ℓ_ν for $|\mathcal{I}_\nu| = |\mathcal{C}_\nu|$.

the gradient of $\ell_\nu(\mathbf{x}_\nu)$ into two components. One component has the same direction as the gradient of $L_\nu(\mathbf{x}_\nu)$, the other component is orthogonal to the gradient of $L_\nu(\mathbf{x}_\nu)$. Thus, minimizing $\ell_\nu(\mathbf{x}_\nu)$ by decreasing along the direction of the steepest descent of $\ell_\nu(\mathbf{x}_\nu)$ means that $L_\nu(\mathbf{x}_\nu)$ is never increased by such a move.

Analysis of Net Model II: Again, we will first discuss the case $|\mathcal{I}_\nu| = 2$. In this case $\hat{\underline{x}}^\alpha = \check{\underline{x}}^\alpha = \underline{x}_\beta$ and $\hat{\underline{x}}^\beta = \check{\underline{x}}^\beta = \underline{x}_\alpha$.

For nets connecting two $\mathcal{I}_\nu$-cells with some $\mathcal{O}_\nu$-cells (Type B) we obtain

$$\ell_\nu(x_\alpha, x_\beta) = \max(\hat{x}, \underline{x}_\beta, x_\alpha) - \min(\check{x}, \underline{x}_\beta, x_\alpha) + \max(\hat{x}, \underline{x}_\alpha, x_\beta) - \min(\check{x}, \underline{x}_\alpha, x_\beta) \quad (35)$$

from (22).

The level contours of the half perimeter and the cost function are illustrated in Fig. 11. In the shaded regions the gradients of the half perimeter and the cost function are identical. These regions are smaller than the shaded regions of net model I. Outside the shaded regions the angle between the gradients is 45° or 90°. The optimum points of $\ell_\nu(x_\alpha, x_\beta)$ contain not only the optimum points of $L_\nu(x_\alpha, x_\beta)$, but also other points. In these points, $\ell_\nu(x_\alpha, x_\beta)$ is not further optimized, although there is some potential to decrease the half perimeter $L_\nu(x_\alpha, x_\beta)$. At first sight these arguments restrict the possibilities of optimizing the half perimeter with net model II and favor net model I. But net model II has one important advantage: The gradients of $L_\nu(x_\alpha, x_\beta)$ and $\ell_\nu(x_\alpha, x_\beta)$ are equal at the point of the current placement $(\underline{x}_\alpha, \underline{x}_\beta)$, which is marked with a • in Fig. 11. Thus, $\mathbf{g}_\ell(x_\alpha, x_\beta)$ and $\mathbf{g}_L(x_\alpha, x_\beta)$ are equal in the neighborhood of the current placement. Since the moves of the cells typically decrease during the iterative improvement process, net model II becomes a more and more accurate approximation of the half perimeter.

If a net connects two $\mathcal{I}_\nu$-cells with no $\mathcal{O}_\nu$-cells (Type C), (22) can be written as

$$\ell_\nu(x_\alpha, x_\beta) = \max(x_\phi, \underline{x}_\beta, x_\alpha) - \min(x_\phi, \underline{x}_\beta, x_\alpha) + \max(x_\phi, \underline{x}_\alpha, x_\beta) - \min(x_\phi, \underline{x}_\alpha, x_\beta). \quad (36)$$

The level contours of the half perimeter and the cost function are illustrated in Fig. 12. For this type, the same arguments hold as for Type B except that the angle between the gradients in the dark shaded regions is 135°. This means that the gradients have opposite components, if both cells lie between x_ϕ and one of the cell coordinates $\underline{x}_\alpha, \underline{x}_\beta$ and $x_\alpha > x_\beta$ holds. But the gradients of $L_\nu(x_\alpha, x_\beta)$ and $\ell_\nu(x_\alpha, x_\beta)$ are again equal at the point of the current placement $(\underline{x}_\alpha, \underline{x}_\beta)$.

Now we will discuss the relation between the gradients of $L_\nu(\mathbf{x}_\nu)$ and $\ell_\nu(\mathbf{x}_\nu)$ in the general case.

Theorem 2: The scalar product of the gradients $\mathbf{g}_L(\mathbf{x}_\nu)$ and $\mathbf{g}_\ell(\mathbf{x}_\nu)$ is always greater or equal 0, if $|\mathcal{I}_\nu| < |\mathcal{C}_\nu|$.

Proof: Since $|\mathcal{I}_\nu| < |\mathcal{C}_\nu|$, it is sufficient to investigate the Types A and B.

Type A ($|\mathcal{I}_\nu| = 1$): The same arguments as in the proof of Theorem 1 for Type A hold here. Thus, $\mathbf{g}_L^T(\mathbf{x}_\nu) \cdot \mathbf{g}_\ell(\mathbf{x}_\nu) \geq 0$.

Type B ($1 < |\mathcal{I}_\nu| < |\mathcal{C}_\nu|$): Suppose a net ν with $\mathcal{I}_\nu = \{\alpha, \ldots, \mu, \ldots, \beta\}$. Without loss of generality we consider the case $x_\alpha < \cdots < x_\mu < \cdots < x_\beta$. All other coordinate orderings can be reduced to this case. $\mathbf{g}_L(\mathbf{x}_\nu) = \mathbf{g}_L(x_\alpha, x_\beta)$ has already been calculated in the proof of Theorem 1 (see (31)).

When calculating $\mathbf{g}_\ell(\mathbf{x}_\nu) = [g_{\ell\alpha}(x_\alpha), \ldots, g_{\ell\mu}(x_\mu), \ldots, g_{\ell\beta}(x_\beta)]^T$ from (22), we get

$$g_{\ell\mu}(x_\mu) = \begin{cases} -1 & \text{if } x_\mu < \min(\check{x}, \check{\underline{x}}^\mu) \\ 0 & \text{if } \min(\check{x}, \check{\underline{x}}^\mu) < x_\mu < \max(\hat{x}, \hat{\underline{x}}^\mu) \\ 1 & \text{if } \max(\hat{x}, \hat{\underline{x}}^\mu) < x_\mu \end{cases} \quad (37)$$

for all $\mu \in \mathcal{I}_\nu$. Simple arithmetic shows for all x_α: $g_{L\alpha}(x_\alpha) \cdot g_{\ell\alpha}(x_\alpha) \geq 0$ and for all x_β: $g_{L\beta}(x_\beta) \cdot g_{\ell\beta}(x_\beta) \geq 0$. Thus, $\mathbf{g}_L^T(x_\alpha, x_\beta) \cdot \mathbf{g}_\ell(\mathbf{x}_\nu) \geq 0$. □

Theorem 3: The gradients $\mathbf{g}_L(\mathbf{x}_\nu)$ and $\mathbf{g}_\ell(\mathbf{x}_\nu)$ are equal at the point $\underline{\mathbf{x}}_\nu$ of the current placement, i.e. $\mathbf{g}_L(\mathbf{x}_\nu = \underline{\mathbf{x}}_\nu) = \mathbf{g}_\ell(\mathbf{x}_\nu = \underline{\mathbf{x}}_\nu)$.

Proof: Again, we will prove this for the Types A,B, and C separately.

Type A($|\mathcal{I}_\nu| = 1$): Since for this type the gradients $\mathbf{g}_L(\mathbf{x}_\nu)$ and $\mathbf{g}_\ell(\mathbf{x}_\nu)$ are identical at all points $\mathbf{x}_\nu$, this holds also at the point $\underline{\mathbf{x}}_\nu$, i.e. $\mathbf{g}_L(\mathbf{x}_\nu = \underline{\mathbf{x}}_\nu) = \mathbf{g}_\ell(\mathbf{x}_\nu = \underline{\mathbf{x}}_\nu)$.

Type B($1 < |\mathcal{I}_\nu| < |\mathcal{C}_\nu|$): Suppose we are given a net ν with $\mathcal{I}_\nu = \{\alpha, \ldots, \mu, \ldots, \beta\}$ and $|\mathcal{I}_\nu| < |\mathcal{C}_\nu|$. As in the proof of Theorem 2 we consider the case $\underline{x}_\alpha < \cdots < \underline{x}_\mu < \cdots < \underline{x}_\beta$. From (31) follows $\mathbf{g}_L(\mathbf{x}_\nu = \underline{\mathbf{x}}_\nu) = [g_{L\alpha}(x_\alpha = \underline{x}_\alpha), 0, \ldots, 0, g_{L\beta}(x_\beta = \underline{x}_\beta)]^T$ with

$$g_{L\alpha}(x_\alpha = \underline{x}_\alpha) = \begin{cases} -1 & \text{if } \underline{x}_\alpha < \check{x} \\ 0 & \text{if } \underline{x}_\alpha > \check{x} \end{cases}$$

$$\text{and } g_{L\beta}(x_\beta = \underline{x}_\beta) = \begin{cases} 0 & \text{if } \underline{x}_\beta < \hat{x} \\ 1 & \text{if } \underline{x}_\beta > \hat{x} \end{cases}. \quad (38)$$

Now, we calculate $\mathbf{g}_\ell(\mathbf{x}_\nu = \underline{\mathbf{x}}_\nu) = [g_{\ell\alpha}(x_\alpha = \underline{x}_\alpha), \ldots, g_{\ell\mu}(x_\mu = \underline{x}_\mu), \ldots, g_{\ell\beta}(x_\beta = \underline{x}_\beta)]^T$. From (37) we get

$$g_{\ell\mu}(x_\mu = \underline{x}_\mu) = \begin{cases} -1 & \text{if } \underline{x}_\mu < \min(\check{x}, \underline{\check{x}}^\mu) \\ 0 & \text{if } \min(\check{x}, \underline{\check{x}}^\mu) < \underline{x}_\mu < \max(\hat{x}, \underline{\hat{x}}^\mu) \\ 1 & \text{if } \max(\hat{x}, \underline{\hat{x}}^\mu) < \underline{x}_\mu \end{cases} \tag{39}$$

for all $\mu \in \mathcal{I}_\nu$. Equation (39) can also be written as:

$$g_{\ell\mu}(x_\mu = \underline{x}_\mu) = \begin{cases} -1 & \text{if } (\underline{x}_\mu < \check{x}) \wedge (\underline{x}_\mu < \underline{\check{x}}_\mu) \\ 0 & \text{if } [(\hat{x} < \underline{x}_\mu) \vee (\underline{\check{x}}^\mu < \underline{x}_\mu)] \wedge [(\underline{x}_\mu < \hat{x}) \vee (\underline{x}_\mu < \underline{\hat{x}}^\mu)] \\ 1 & \text{if } (\hat{x} < \underline{x}_\mu) \wedge (\underline{\hat{x}}^\mu < \underline{x}_\mu) \end{cases} \tag{40}$$

Since $\underline{x}_\alpha < \cdots < \underline{x}_\mu < \cdots < \underline{x}_\beta$, it follows

$$\begin{aligned} \underline{x}_\alpha &< \underline{\check{x}}^\alpha < \underline{\hat{x}}^\alpha, \\ \underline{\check{x}}^\mu &< \underline{x}_\mu < \underline{\hat{x}}^\mu \ (\mu \in \mathcal{I}_\nu \backslash \{\alpha, \beta\}), \text{ and} \\ \underline{\check{x}}^\beta &< \underline{\hat{x}}^\beta < \underline{x}_\beta. \end{aligned} \tag{41}$$

With these inequalities we obtain the components of the gradient $\mathbf{g}_\ell(\mathbf{x}_\nu = \underline{\mathbf{x}}_\nu)$ from (40):

$$\begin{aligned} g_{\ell\alpha}(x_\alpha = \underline{x}_\alpha) &= \begin{cases} -1 & \text{if } \underline{x}_\alpha < \check{x} \\ 0 & \text{if } \underline{x}_\alpha > \check{x} \end{cases}, \\ g_{\ell\beta}(x_\beta = \underline{x}_\beta) &= \begin{cases} 0 & \text{if } \underline{x}_\beta < \hat{x} \\ 1 & \text{if } \underline{x}_\beta > \hat{x} \end{cases}, \end{aligned} \tag{42}$$

and $g_{\ell\mu}(x_\mu = \underline{x}_\mu) = 0$ for all $\mu \in \mathcal{I}_\nu \backslash \{\alpha, \beta\}$.
After comparing the components of $\mathbf{g}_\ell(\mathbf{x}_\nu = \underline{\mathbf{x}}_\nu)$ and $\mathbf{g}_L(\mathbf{x}_\nu = \underline{\mathbf{x}}_\nu)$, we see that $\mathbf{g}_\ell(\mathbf{x}_\nu = \underline{\mathbf{x}}_\nu) = \mathbf{g}_L(\mathbf{x}_\nu = \underline{\mathbf{x}}_\nu)$.

Type $C(|\mathcal{I}_\nu| = |\mathcal{C}_\nu|)$: For nets of this type (20) yields: $\mathbf{g}_L(\mathbf{x}_\nu = \underline{\mathbf{x}}_\nu) = [g_{L\alpha}(x_\alpha = \underline{x}_\alpha), 0, \ldots, 0, g_{L\beta}(x_\beta = \underline{x}_\beta)]^T$ with $g_{L\alpha}(x_\alpha = \underline{x}_\alpha) = -1$ and $g_{L\beta}(x_\beta = \underline{x}_\beta) = 1$.

When calculating $\mathbf{g}_\ell(\mathbf{x}_\nu = \underline{\mathbf{x}}_\nu) = [g_{\ell\alpha}(x_\alpha = \underline{x}_\alpha), \ldots, g_{\ell\mu}(x_\mu = \underline{x}_\mu), \ldots, g_{\ell\beta}(x_\beta = \underline{x}_\beta)]^T$ we set $\hat{x} = \check{x} = x_\phi$ in (40). This yields:

$$g_{\ell\mu}(x_\mu = \underline{x}_\mu) = \begin{cases} -1 & \text{if } (\underline{x}_\mu < x_\phi) \wedge (\underline{x}_\mu < \underline{\check{x}}^\mu) \\ 0 & \text{if } [(x_\phi < \underline{\check{x}}_\mu) \vee (\underline{\check{x}}^\mu < \underline{x}_\mu)] \wedge [(\underline{x}_\mu < x_\phi) \vee (\underline{x}_\mu < \underline{\hat{x}}^\mu)] \\ 1 & \text{if } (x_\phi < \underline{x}_\mu) \wedge (\underline{\hat{x}}^\mu < \underline{x}_\mu) \end{cases}. \tag{43}$$

Since the virtual cell ϕ is positioned in the center of gravity of all $\mathcal{I}_\nu$-cells connected by a net, we get

$$\underline{x}_\alpha < x_\phi < \underline{x}_\beta. \tag{44}$$

With this inequality and the inequalities in (41), (43) yields: $g_{\ell_\alpha}(x_\alpha = \underline{x}_\alpha) = -1$, $g_{\ell_\mu}(x_\mu = \underline{x}_\mu) = 0$ for all $\mu \in \mathcal{I}_\nu \backslash \{\alpha, \beta\}$, and $g_{\ell\beta}(x_\beta = \underline{x}_\beta) = 1$. When comparing again the components of the gradients, we get $\mathbf{g}_L(\mathbf{x}_\nu = \underline{\mathbf{x}}_\nu) = \mathbf{g}_\ell(\mathbf{x}_\nu = \underline{\mathbf{x}}_\nu) = [-1, 0, \ldots, 0, 1]^T$. □

TABLE I
CHARACTERISTICS OF BENCHMARK EXAMPLES

circuit	#cells	#pads	#nets	#pins	#rows
primary1	752	81	904	2941	20
struct	1888	64	1920	5471	19
primary2	2907	107	3029	11229	28
biomed	6417	97	5742	21040	39
industry2	12142	495	13419	48404	66
industry3	15032	68	21924	68290	60
avq.small	21854	64	22120	60729	80
avq.large	25114	64	25378	65578	72

TABLE II
RESULTS (AREA IN mm^2, CPU TIME IN *SECONDS*)

circuit	TimberWolf SC 5.4		VPNR (cprt)		GORDIANL		GORDIANL& DOMINOI		GORDIANL& DOMINOII	
	area	cpu	area	cpu	area	cpu	area	cpu	area	cpu
primary1	20.31	794	23.37	99	20.76	81	19.96	151	20.03	168
struct	6.56	2875	8.23	174	6.48	167	6.38	245	6.20	253
primary2	78.16	5316	95.33	580	78.85	496	78.58	838	78.11	922
biomed	48.22	14631	53.55	5567	45.93	1639	40.43	2742	39.99	2640
industry2	219.52	37521	272.40	16399	233.59	7550	215.61	9835	214.25	9587
industry3	658.01	65652	755.07	11571	598.42	8458	586.92	10257	575.54	10349
avq.small	161.05	92959	n.a.		129.39	14488	123.57	15963	122.28	17444
avq.large	168.07	96564	n.a.		141.38	18537	130.10	20549	129.27	21086
average	1.00	1.00	1.18	0.21	0.95	0.13	0.90	0.18	0.89	0.18

IV. EXPERIMENTAL RESULTS

DOMINO results for net models I and II are compared with the placement results of several well-known tools that were made available to us. We tested TimberWolfSC 5.4 [16], the VPNR cplrt (combined place and route) algorithm [4], and GORDIANL [11] on the basis of the benchmark examples from [30]. TimberWolf is the best known placement algorithm and it is still further improved. The characteristics of the benchmark circuits containing approximately 800 to 25,000 cells are summarized in Table I. The placement methods are compared based on layout areas after final routing given in Table II. All routing areas have been determined with the TimberWolf global router [31] and the VPNR final routing package [32]. For each method the computation times required for placement are also included. The DOMINO times include the times required for the GORDIANL initial placement. The experiments were conducted using a DECstation 5000/200 with 128 MB of main memory.

The results show that TimberWolf produces placements better in quality than the VPNR cplrt algorithm. TimberWolf obtained 18% smaller layout areas taking five times the computation time on the average. GORDIANL produces placements better in quality than TimberWolfSC 5.4. GORDIANL obtained 5% smaller layout areas taking about eight times less computation time on the average. The areas obtained by DOMINO with net model II are always smaller compared to net model I except for circuit *primary1*. DOMINO with net model II yields the best results with 11% smaller layout areas than TimberWolf on the average.

These results can be obtained with less computation time than TimberWolf. For example, the circuit *avq. large* with 25,000 cells has been placed in 5.8 hours. The additional computation time needed for the iterative DOMINO procedure is only moderate with respect to the initial placement procedure GORDIANL.

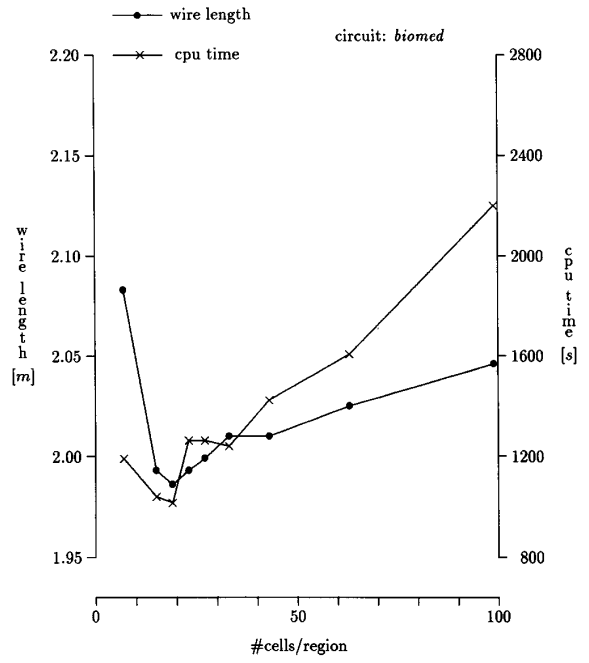


Fig. 13. Net model I: Impact of #cells/regions.

At the 1992 MCNC International Workshop on Layout Synthesis the most recent placement contest, called TimberWolf Hunt, was held. TimberWolf and GORDIAN L & DOMINOI were compared on a real design of 13,770 cells and 16,642 nets. The example was not revealed to the participants before the contest so that circuit specific tuning was prohibited. The quality of the placement solution of GORDIANL & DOMINO I was slightly better using one fifth the cpu-time to execute (see Table III).

For the very large circuit *golem* [33] with about 100,000 cells the length of the minimum spanning trees of the placement obtained with GORDIANL & DOMINOII is 22% smaller compared to TimberWolf (see Table IV).

For all benchmark circuits, the number of generations is between 2 and 10 where the number increases with circuit size.

We conducted an experiment to identify the optimal region size of the DOMINO procedure. The circuit *biomed* was used for this experiment. Figs. 13 and 14 show the impact of the number of cells per region on the cpu time and the wire length for net model I and net model II, respectively. The wire length is measured by the half perimeter.

The cpu time increases along with the #cells/region because of the cubic time complexity of the transportation algorithm. The larger cpu time with 5 cells/region than with 20 cells/region results from the larger number of generations needed by the iterative process to converge with 5 cells/region.

With a small #cells/region the cells were moved only small distances, and the wire length of the placement obtained with DOMINO is high. The wire length decreases along with an increasing #cells/region, because more cells are placed simultaneously. In Fig. 13 the half perimeter increases for large regions, since the number of nets of Type C increases.

TABLE III
TIMBERWOLF HUNT CIRCUIT (AREA IN mm^2, CPU TIME IN *SECONDS*)

				TimberWolfSC		GORDIANL & DOMINOI	
circuit	#cells	#pads	#nets	area	cpu	area	cpu
TimberWolf Hunt	13770	76	16642	13.28	53044	13.21	10391
				1.00	1.00	0.99	0.19

TABLE IV
GOLEM CIRCUIT (MINIMUM SPANNING TREES (MST) IN UNITS, CPU TIME IN *SECONDS*)

				TimberWolfSC 5.4		GORDIANL & DOMINOII	
circuit	#cells	#pads	#nets	MST	cpu	MST	cpu
golem	99932	380	144949	$2.10 \cdot 10^7$	1011387	$1.65 \cdot 10^7$	132090
				1.00	1.00	0.78	0.13

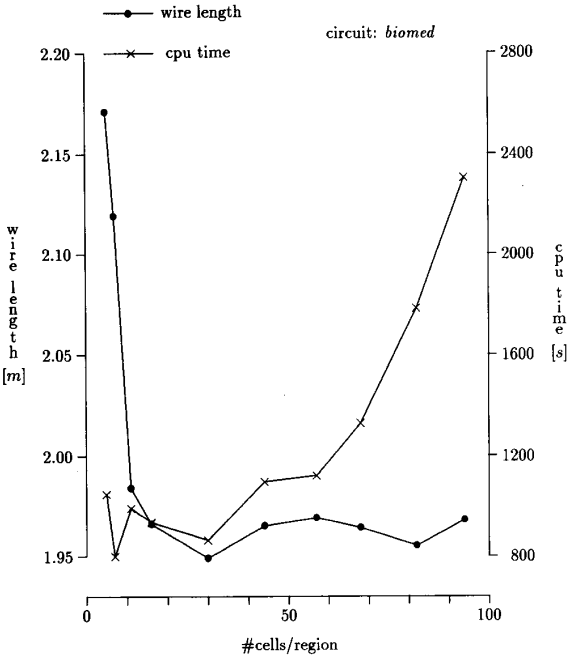


Fig. 14. Net model II: Impact of #cells/regions.

This results in an increasing angle φ between the gradients of the half perimeter and the cost function (see (34)) and, therefore, in a less accurate approximation of the half perimeter by the cost function. For net model II (Fig. 14) nearly the same results are obtained for 20 and more cells per region. It should be noted that wire length does not increase, because the gradients of the half perimeter and the cost function are equal in the point of the current placement. This property does not depend on the number of cells per region. The curves in Figs. 13 and 14 show that a #cells/region of about 20 to 30 yields good layout quality as well as short cpu-time.

These results indicate that the transportation algorithm combined with the presented net models is an appropriate and efficient method for cell placement.

V. CONCLUSION

We have described an effective and efficient interative improvement method for the placement of cells. In each iterative step the rearrangement of cells of different sizes

is formulated as a transportation problem that is solved by a network flow algorithm. To determine the transportation costs, two new net models were introduced. The relations between the new net models and the half perimeter have been theoretically analyzed. This explains why DOMINO's results are superior to other placement methods.

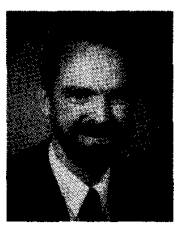
ACKNOWLEDGMENT

The authors would like to acknowledge G. Sigl for his valuable suggestions. They thank F. Brglez and K. Kozminski from MCNC for providing the VPNR-package, Prof. C. Sechen for making TimberWolfSC 5.4 available, and the anonymous reviewers for their helpful comments. Many discussions with Prof. J. Cohoon during his sabbatical in Munich greatly improved this paper.

REFERENCES

[1] B. Preas and M. Lorenzetti, *Physical Design Automation of VLSI Systems*. Benjamin/Cummings, 1988.
[2] T. Lengauer, *Combinatorial Algorithms for Integrated Circuit Layout*. John Wiley & Sons, 1990.
[3] U. Lauther, "A min-cut placement algorithm for general cell assemblies based on a graph representation," in *ACM/IEEE Proc. 16th Design Automation Conf.*, 1979, pp. 1–10.
[4] P. Suaris and G. Kedem, "An algorithm for quadrisection and its application to standard cell placement," *IEEE Trans. Circuits Syst.*, vol. CAS-35, pp. 294–303, Mar. 1988.
[5] R. H. J. M. Otten, "Eigensolutions in top-down layout design," in *Proc. IEEE Int. Symp. on Circuits and Systems*, 1982, pp. 1017–1020.
[6] J. P. Blanks, "Near-optimal placement using a quadratic objective function," in *Proc. ACM/IEEE 22th Design Automation Conf.*, 1985, pp. 609–615.
[7] L. Sha and R. W. Dutton, "An analytical algorithm for placement of arbitrarily sized rectangular blocks," in *Proc. ACM/IEEE 22nd Design Automation Conf.*, 1985, pp. 602–608.
[8] R.-S. Tsay, E. S. Kuh, and C.-P. Hsu, "Proud: A fast sea-of-gates placement algorithm," in *Proc. ACM/IEEE 25th Design Automation Conf.*, 1988, pp. 318–323.
[9] J. M. Kleinhans, G. Sigl, and F. M. Johannes, "Gordian: A new global optimization/rectangle dissection method for cell placement," in *Proc. IEEE Int. Conf. Computer-Aided Design*, 1988, pp. 506–509.
[10] J. Kleinhans, G. Sigl, F. Johannes, and K. Antreich, "Gordian: VLSI placement by quadratic programming and slicing optimization," *IEEE Trans. Computer-Aided Design*, vol. CAD-10, pp. 356–365, Mar. 1991.
[11] G. Sigl, K. Doll, and F. Johannes, "Analytical placement: A linear or a quadratic objective function?," in *Proc. ACM/IEEE 28th Design Automation Conf.*, 1991, pp. 427–432.
[12] R. Kling and P. Banerjee, "ESP: Placement by simulated evolution," *IEEE Trans. Computer-Aided Design*, vol. CAD-8, pp. 245–256, Mar. 1989.
[13] J. P. Cohoon and W. D. Paris, "Genetic placement," in *Proc. IEEE Int. Conf. Computer-Aided Design*, 1986, pp. 374–377.
[14] S. Kirkpatrick, J. C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, pp. 671–680, Feb. 1982.
[15] F. I. Romeo, *Simulated Annealing: Theory and Applications to Layout Problems*, Electronics Research Laboratory, University of California-Berkely, Memorandum No. UCB/ERL M89/29, Mar. 1989.
[16] C. Sechen and K. W. Lee, "An improved simulated annealing algorithm," in *Proc. IEEE Int. Conf. Computer-Aided Design*, pp. 478–481, 1987.
[17] M. Hanan, P. K. Wolff, and B. J. Agule, "Some experimental results on placement techniques," in *Proc. ACM/IEEE 13th Design Automation Conf.*, 1976, pp. 214–224.
[18] N. R. Quinn and M. A. Breuer, "A force directed component placement procedure for printed circuit boards," *IEEE Trans. Circuits Syst.*, vol. CAS-26, pp. 377–388, June 1979.
[19] S. Goto and E. S. Kuh, "An approach to the two-dimensional placement problem in circuit layout," *IEEE Trans. Circuits Syst.*, vol. CAS-25, pp. 208–214, Apr. 1978.
[20] L. Steinberg, "The backboard wiring problem: A placement algorithm," *SIAM Review*, pp. 37–50, 1961.
[21] H. W. Carter, M. A. Breuer, and Z. A. Syed, "Incremental processing applied to Steinberg's placement procedure," in *Proc. ACM/IEEE 16th Design Automation Conf.*, 1979, pp. 26–31.
[22] S. B. Akers, "On the use of the linear assignment algorithm in module placement," in *Proc. ACM/IEEE 18th Design Automation Conf.*, 1981, pp. 137–144.
[23] A. Srinivasan, K. Chaudhary, and E. S. Kuh, "Ritual: A performance driven placement algorithm for small cell ICs," in *Proc. IEEE Int. Conf. Computer-Aided Design*, 1991, pp. 48–51.
[24] K. Doll, F. Johannes, and G. Sigl, "Domino: Deterministic placement improvement with hill-climbing capabilities," in *Proc. IFIP TC 10/WG 10.5 Int. Conf. on Very Large Scale Integration*, 1991, pp. 91–100.
[25] K. Doll, F. Johannes, and G. Sigl, "Placement improvement by network flow methods," in *Proc. International Workshop on Layout Synthesis*, 1982, pp. 179–188.
[26] K. Doll, F. Johannes, and G. Sigl, "Accurate net models for placement improvement by network flow methods," in *Proc. IEEE Int. Conf. Computer-Aided Design*, 1992, pp. 594–597.
[27] H. Shin, A. Sangiovanni-Vincentelli, and C. Sequin, "Zone-refining techinques for IC layout compaction," *IEEE Trans. Computer-Aided Design*, vol. CAD-9, pp. 167–179, Feb. 1990.
[28] J. L. Ford and D. Fulkerson, *Flows In Networks*. Princeton, New Jersey: Princeton University Press, 1962.
[29] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*. London: Academic Press. Inc., 1981.
[30] K. Kozminski, "Benchmarks for layout synthesis," in *Proc. ACM/IEEE 28th Design Automation Conf.*, 1991, pp. 265–270.
[31] K. Lee and C. Sechen, "A new global router for row-based layout," in *Proc. IEEE Int. Conf. Computer-Aided Design*, 1988, pp. 180–183.
[32] "VPNR users guide," Microelectronics Center of North Carolina, MCNC Tech. Rep., 1988.
[33] J. Koehl, F. Braun, and R.-S. Tsay, "Golem large chip testcase: Data format," IBM Tech. Rep., 1992.

Konrad Doll received the Dipl.-Ing. degree in electrical engineering from the Technical University of Munich, Germany, in 1989. From June 1989 to March 1994 he worked towards the Dr.-Ing. degree at the Institute of Electronic Design Automation, Technical University of Munich, where he was a research assistant. He is currently with Motorola GmbH, Munich, Germany. His research interests extend to all aspects of computer-aided design of integrated circuits, in particular, automatic layout of digital circuits including partitioning, floor-planning, placement, and routing.



Frank M. Johannes received the Dipl.-Ing. degree in electrical engineering from the Technical University of Karlsruhe, Karlsruhe, Germany, in 1968, and the Dr.-Ing. degree from the University of Erlangen-Nürnberg, Erlangen, Germany, in 1973. From 1968 to 1975, he was with the Regional Computing Center in Erlangen, working in the fields of computer networking and computer-aided software engineering. In 1976, he joined the Institute of Electronic Design Automation, Department of Electrical Engineering, Technical University of Munich, Munich, Germany. Since 1993, he has been a Professor of Electrical Engineering at the Technical University of Munich. His research interests are in the area of computer-aided design of electronic circuits and systems, at present with emphasis on layout synthesis. Dr. Johannes is a member of the ACM.

Kurt J. Antreich (M'69–SM'78–F'94) received the Dipl.-Ing. degree from the Technical University of Munich, Munich, Germany, in 1959, and the Dr.-Ing. degree from the Technical University Fridericiana Karlsruhe, Karlsruhe, Germany, in 1966. He joined AEG-Telefunken in 1959, where he was involved in computer-aided circuit design for telecommunication systems. From 1968 to 1975 he was head of the Advanced Development Department of AEG-Telefunken, Backnang, Germany. Since 1975, he has been a professor of electrical engineering at the Technical University of Munich, Munich, Germany. His research interests are in computer-aided design of electronic circuits and systems, with particular emphasis on circuit optimization, testing, layout synthesis, and synthesis of digital systems. He was Chairman of the ITG Circuit and Systems Group from 1972 to 1974, and a member of the executive board of the ITG from 1979 to 1982, and Chairman of the ITG CAD Group from 1985 to 1989. He received the ITG prize paper award in 1976. Dr. Antreich is a member of the ITG, the GI (Gesellschaft für Informatik, Germany), and the IEEE. (For a photograph of K. J. Antreich, see page 71 of the January 1994 issue of this TRANSACTIONS.)