Orientation assignment of standard cells using a fuzzy mathematical transformation

J.-T. Yan
P.-Y. Hsiao

Abstract: It is well known that minimising total wire length reduces routing area in a standard cell layout. After the placement phase, another advanced improvement on total wire length is made by assigning the orientations of standard cells. In the paper, based on fuzzy c-means clustering, the authors develop two-way constrained fuzzy graph clustering and the transformation between the orientation assignment of standard cells and the constrained graph bisection to minimise total wire length. Finally, the proposed algorithm has tested several standard cell layouts, and the experimental results show that the proposed algorithm produces significant wire reduction on total wire length.

1 Introduction

In VLSI physical design, the main object of the placement phase is to place a given set of circuit modules on the chip surface with minimum area or total wire length. For most of the placement approaches [1] the centres of circuit modules are usually considered as the circuit bodies during the placement phase. Thus all the nets between the circuit modules must be constructed by connecting these centres. Owing to different estimation methods of wire length, the placement results are obtained by different placement algorithms. In general, all the modules are placed on the chip surface and the pins on each module are assigned to the boundary of the module to be applied to the routing phase after the placement phase. Furthermore, it is well known that minimising total wire length will reduce the routing area in a circuit module layout. Hence, one advanced improvement [2] on total wire length is made by flipping these fixed modules with respect to its vertical and/or horizontal axes of symmetry or rotating the fixed modules by 0, 90, 180, 270 degrees before the routing phase. In Fig. 1, the different orientations of a circuit module and the transform diagram between the different orientations is illustrated. In general, the orientation assignment of the modules, called the orientation problem, is to minimise total wire length by further assigning orientations of modules. On the other hand, the rotation assignment of the modules, called the rotation problem, is to minimise total wire length by further assigning rotations of modules.
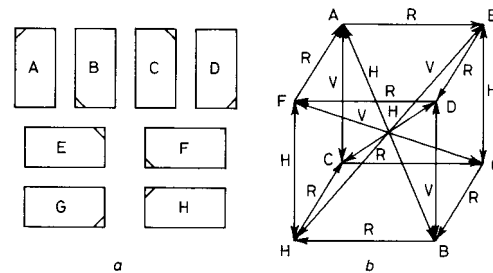
Fig. 1
a Different orientations of a circuit module
b The transform diagram for different orientations
R = rotate 90 degree; H = horizontal flip; V = vertical flip

For the orientation problem, the flipping operation was adopted in standard-cell and gate-array style layout systems [3] in 1977. In a macro cell layout, an analytical method was proposed to solve the orientation problem by M. Yamada and C.L. Liu [4], and several testing examples were generated and studied. However, the cost function for total wire length is the sum of the squares of the wire lengths. The experimental results of these examples cannot be compared with other approaches in which the cost function is in a different form. In 1989, a neural-based method was further proposed by R.L. Hadas and C.L. Liu [5]. The orientation problem for a macro cell layout is proven to be NP-complete. It was reported that the two methods based on simulated annealing and neural computation networks produce very good experimental results. Unfortunately, the computation time of the two methods is very long for a large problem. For the consideration of computation complexity, a fast heuristic algorithm was proposed by X. Yao and C.L. Liu [6]. Furthermore, using a graph partitioning technique, the problem was also proven to be NP-complete, and a graph-based algorithm was proposed in $O(n^2 \log n)$ by Cheng, Hu and Yao [7]. In 1991, Jeong and Kyung proposed an integer programming approach [8] to find optimal module orientations in a macro cell layout. The orientation problem for a standard cell layout and a matrix of macro cells is also discussed by Chong and Sahni [9]. Polynomial time algorithms were proposed, and it was shown that a simple greedy heuristic often outperforms the neural network and simulated annealing heuristics.

In this paper, first, based on the hypothesis of the Manhattan net model, the orientation problem [15] in a standard cell layout can be divided into the VO problem

and the HO problem. Furthermore, according to the physical property in a standard cell layout, the VO problem will be optimally solved in $O(n)$ time. On the other hand, based on fuzzy c-means clustering [10, 11], we develop two-way constrained fuzzy graph clustering and the transformation between the orientation assignment of standard cells and the constrained graph bisection to minimise total wire length for the HO problem. The transformation of the orientation problem based on the Manhattan net model and two-way constrained fuzzy graph clustering is shown in Fig. 2. Finally, the proposed
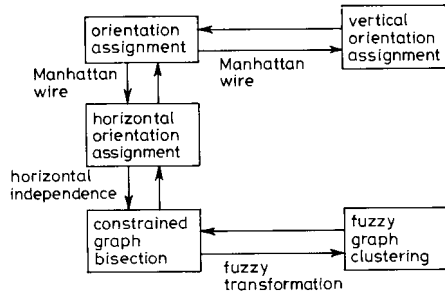

Fig. 2   *Problem transformation of orientation problem*

algorithm has tested several standard cell layouts, and the experimental results show that the proposed algorithm produces significant wire reduction on total wire length.

2   Problem description and preliminaries

For a standard cell layout [12], the cells are positioned on a fixed chip surface row by row. The cells in one row have the same height and are arranged in a good fixed order. The pins are only located on top or bottom boundaries of cells, and the cells are connected by routing the nets among the pins of standard cells. In the orientation problem for a standard cell layout, the problem is to minimise total wire length or maximise wire reduction by further assigning vertical and horizontal orientations for each cell. In Fig. 3 a standard cell layout is shown.
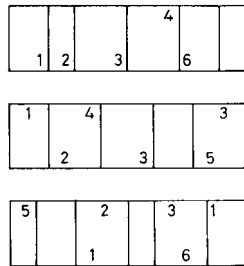

Fig. 3   *Standard cell layout*

2.1   Wire length estimation

For the orientation problem, the objective function is to minimise total wire length in a standard cell layout. Hence, to estimate the wire length of a net becomes important for computing total wire length in a standard cell layout. In general, the Manhattan path is usually applied to model the wire behaviour of a net in the routing phase. Thus, for any connecting net, we assume that the minimum Manhattan distance of a net is the

wire length of the net. For any two connecting points located at (x, y) and (u, v), the minimum Manhattan distance is obtained as $|x - u| + |y - v|$. In general, the nets can be divided into two-pin nets and multiple-pin nets. For a two-pin net N whose pins are (x_1, y_1) and (x_2, y_2), the wire length will be measured as Len (N) $= |x_1 - x_2| + |y_1 - y_2|$. On the other hand, for one multipin net N, the minimum Manhattan distance of the net is the length of the minimum rectilinear Steiner tree of the net. It is well known that the minimum rectilinear Steiner tree problem [13] is an NP-complete problem. Hence, it is further assumed that the wire length of the net is the length of the minimum rectilinear spanning tree (MRST) of the net. A multiple-pin net N will be decomposed into several two-pin nets using an MRST model [14], and the wire length is the sum of the wire lengths of the decomposed two-pin nets, such as

$$\text{Len (N)} = \sum_{(i, j) \in N_{MRST}} \{ |x_i - x_j| + |y_i - y_j| \}$$

where N_{MRST} is a set of two-pin nets using an MRST model for a multiple-pin net N. The nets in Fig. 3 are routed by an MRST model and shown in Fig. 4.
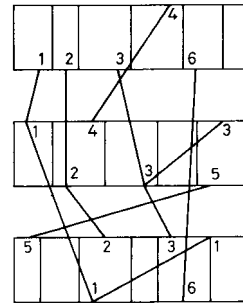

Fig. 4   *Estimation of routing nets by MRST model*

2.2   Orientation problem

Formally speaking, for the orientation problem, each standard cell S has an orientation state (p_s, q_s), $p_s, q_s \in \{0, 1\}$, and the states $(p_s, q_s) = (0, 0)$, $(1, 0)$, $(0, 1)$ and $(1, 1)$, respectively, denote no flip, one horizontal flip, one vertical flip and both horizontal and vertical flips, respectively, for cell S. Let pin (x, y) locate on a standard cell S whose centre is (x_s, y_s); the co-ordinates $(x, 2y_s - y)$ and $(2x_s - x, y)$, respectively, are the positions of the pin (x, y) by a vertical flip and a horizontal flip, respectively. Thus, the co-ordinate of the pin (x, y) can be generalised as $(x', y') = (x + 2p_s(x_s - x), y + 2q_s(y_s - y))$, $p_s, q_s \in \{0, 1\}$. In Fig. 5, the co-ordinate change by a horizontal or vertical flip is shown.
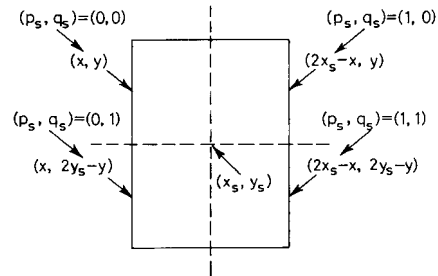

Fig. 5   *Position change by a horizontal or vertical flip*

Owing to the assumptions of Manhattan distance and MRST net model, it is clear that any horizontal (vertical) flip only changes the horizontal (vertical) wire length of the nets in a standard cell layout. Therefore, the orientation problem will be further divided into two independent subproblems: the horizontal orientation (HO) problem and the vertical orientation (VO) problem. For the HO problem, the co-ordinate of the pin (x, y) will be generalised as $(x', y') = (x + 2p_s(x_s - x), y)$, $p_s \in \{0, 1\}$. On the other hand, for the VO problem, the co-ordinate of the pin (x, y) will be generalised as $(x', y') = (x, y + 2q_s(y_s - y))$, $q_s \in \{0, 1\}$. Let a set of two-pin nets in a standard cell layout be defined as $\{1, 2, \ldots, m\}$ and a set of standard cells be defined as $\{1, 2, \ldots, n\}$. By the generalised co-ordinate form of any pin, the total wire length $L_{gen}(\text{HO})$ and $L_{gen}(\text{VO})$ in a standard cell layout for the HO problem and the VO problem are computed as

$$L_{gen}(\text{HO}) = \sum_{(i_s, j_t) \in S_{net}} \{ | x_i + 2p_s(x_s - x_i) - x_j$$
$$- 2p_t(x_t - x_j) | + | y_i - y_j | \}$$
$$p_1, p_2, \ldots, p_n \in \{0, 1\}$$

$$L_{gen}(\text{VO}) = \sum_{(i_s, j_t) \in S_{net}} \{ | x_i - x_j | + | y_i + 2q_s(y_s - y_i)$$
$$- y_j - 2q_t(y_t - y_j) | \}$$
$$q_1, q_2, \ldots, q_n \in \{0, 1\}$$

where (i_s, j_t) represents one two-pin net whose pin $i_s = (x_i + 2p_s(x_s - x_i), y_i + 2q_s(y_s - y_i))$ and $j_t = (x_j + 2p_t(x_t - x_j), y_j + 2q_t(y_t - y_j))$ are located on cell S and T, respectively, and S_{net} is a set of two-pin nets after the transformation of the MRST net model.

From another viewpoint of minimising total wire length for the orientation problem, it is clear that minimising total wire length corresponds to maximising total wire reduction in a standard cell layout. Hence, the objective functions in a standard cell layout for the HO problem and the VO problem will be modelled as maximising $L_{red}(\text{HO})$ and $L_{red}(\text{VO})$:

$$L_{red}(\text{HO}) = \sum_{(i_s, j_t) \in S_{net}} \{ | x_i - x_j | - | x_i + 2p_s(x_s - x_i)$$
$$- x_j - 2p_t(x_t - x_j) | \}$$
$$p_1, p_2, \ldots, p_n \in \{0, 1\}$$

$$L_{red}(\text{VO}) = \sum_{(i_s, j_t) \in S_{net}} \{ | y_i - y_j | - | y_i + 2q_s(y_s - y_i)$$
$$- y_j - 2q_t(y_t - y_j) | \}$$
$$q_1, q_2, \ldots, q_n \in \{0, 1\}$$

In addition to the flipping direction, the HO problem and the VO problem are the same problem. The following definitions are applied to solve the HO problem and the VO problem at the same time. Given a two-pin net N_i, the pins (x_s^i, y_s^i) and (x_t^i, y_t^i) are located on cell S and T whose centres are (x_s, y_s) and (x_t, y_t), respectively.

Definition 1: If cell S is flipped by a horizontal orientation state p_s and cell T is flipped by a horizontal orientation state p_t, the horizontal net reduction $NR_{hor}(\text{S}(p_s), \text{T}(p_t), N_i)$ for N_i is defined as (see Fig. 6)

$$NR_{hor}(\text{S}(p_s), \text{T}(p_t), N_i)$$
$$= | x_t^i - x_s^i | + | y_t^i - y_s^i |$$
$$- | x_t^i + 2p_t(x_t - x_t^i) - x_s^i - 2p_s(x_s - x_s^i) |$$
$$- | y_t^i y - y_s^i |$$
$$= | x_t^i - x_s^i | - | x_t^i + 2p_t(x_t - x_t^i) - x_s^i - 2p_s(x_s - x_s^i) |$$

Furthermore, the horizontal flip reduction $FR_{hor}(\text{S}(p_s), \text{T}(p_t))$ for cell S and T is defined as

$$FR_{hor}(\text{S}(p_s), \text{T}(p_t)) = \sum_{N_i \in (S, T)} NR_{hor}(\text{S}(p_s), \text{T}(p_t), N_i)$$
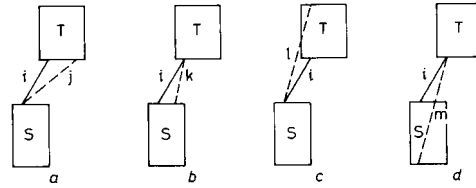
Fig. 6 *Net reduction*

a $NR_{hor}(\text{S}(0), \text{T}(1), i) = \text{Len}\,(i) - \text{Len}\,(j)$
b $NR_{hor}(\text{S}(1), \text{T}(0), i) = \text{Len}\,(i) - \text{Len}\,(k)$
c $NR_{ver}(\text{S}(0), \text{T}(1), i) = \text{Len}\,(i) - \text{Len}\,(l)$
d $NR_{ver}(\text{S}(1), \text{T}(0), i) = \text{Len}\,(i) - \text{Len}\,(m)$

where (S, T) is a set of two-pin nets whose pins are located on cell S and T, respectively. By similar definitions, $NR_{ver}(\text{S}(q_s), \text{T}(q_t), N_i)$ and $FR_{ver}(\text{S}(q_s), \text{T}(q_t))$ are defined as

$$NR_{ver}(\text{S}(q_s), \text{T}(q_t), N_i)$$
$$= | y_t^i - y_s^i | - | y_t^i + 2q_t(y_t - y_t^i) - y_s^i - 2q_s(y_t - y_s^i) |$$

and

$$FR_{ver}(\text{S}(q_s), \text{T}(q_t)) = \sum_{N_i \in (S, T)} NR_{ver}(\text{S}(q_s), \text{T}(q_t), N_i)$$

Definition 2: After cell S has been horizontally flipped, the co-ordinate of the pin (x_s^i, y_s^i) in the net N_i is modified as $(2x_s - x_s^i, y_s^i)$. If cell T is further flipped by a horizontal flip, the virtual horizontal net reduction $VNR_{hor}(\text{S}, \text{T}, N_i)$ for net N_i is defined as (see Fig. 7)

$$VNR_{hor}(\text{S}, \text{T}, N_i)$$
$$= NR_{hor}(\text{S}(1), \text{T}(1), N_i) - NR_{hor}(\text{S}(1), \text{T}(0), N_i)$$
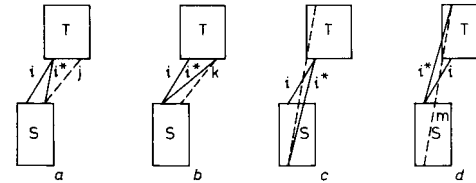
Fig. 7 *Virtual net reduction*

a $VNR_{hor}(\text{S}, \text{T}, i) = \text{Len}\,(i^*) - \text{Len}(j)$
b $VNR_{hor}(\text{T}, \text{S}, i) = \text{Len}\,(i^*) - \text{Len}\,(k)$
c $VNR_{ver}(\text{S}, \text{T}, i) = \text{Len}\,(i^*) - \text{Len}\,(l)$
d $VNR_{ver}(\text{T}, \text{S}, i) = \text{Len}\,(i^*) - \text{Len}\,(m)$

By similar definitions, $VNR_{hor}(\text{T}, \text{S}, N_i)$, $VNR_{ver}(\text{S}, \text{T}, N_i)$ and $VNR_{ver}(\text{T}, \text{S}, N_i)$ are defined as:

$$VNR_{hor}(\text{T}, \text{S}, N_i)$$
$$= NR_{hor}(\text{S}(1), \text{T}(1), N_i) - NR_{hor}(\text{S}(0), \text{T}(1), N_i)$$

$$VNR_{ver}(\text{S}, \text{T}, N_i)$$
$$= NR_{ver}(\text{S}(1), \text{T}(1), N_i) - NR_{ver}(\text{S}(1), \text{T}(0), N_i)$$

$$VNR_{ver}(\text{T}, \text{S}, N_i)$$
$$NR_{ver}(\text{S}(1), \text{T}(1), N_i) - NR_{ver}(\text{S}(0), \text{T}(1), N_i)$$

Definition 3: If $NR_{hor}(\text{S}(0), \text{T}(1), N_i) = VNR_{hor}(\text{S}, \text{T}, N_i)$ and $NR_{hor}(\text{S}(1), \text{T}(0), N_i) = VNR_{hor}(\text{T}, \text{S}, N_i)$, for each $N_i \in (S, T)$, cell S and T will be horizontally independent. On the other hand, if $NR_{ver}(\text{S}(0), \text{T}(1), N_i) = VNR_{ver}(\text{S}, \text{T},$

159

N_i) and $NR_{ver}(S(1), T(0), N_i) = VNR_{ver}(T, S, N_i)$, for each $N_i \in (S, T)$, cells S and T will be vertically independent (see Fig. 8).

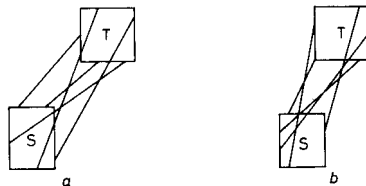*Lemma 1:* Any two standard cells are vertically independent.

Fig. 8 *Cell independennce*
a Horizonal cell independence
b Vertical cell independence

Proof: For a standard cell layout, the pins are only located on top or bottom boundaries of cells, so two pins on any net will be located on top or bottom boundaries of cells. For any net N_i on cells S and T, it is clear that

$$NR_{ver}(S(0), T(1), N_i) = VNR_{ver}(S, T, N_i)$$

$$= \text{height of standard cell T}$$

$$\text{or} \; - (\text{height of standard cell T})$$

and

$$NR_{ver}(T(0), S(1), N_i) = VNR_{ver}(T, S, N_i)$$

$$= \text{height of standard cell S}$$

$$\text{or} \; - (\text{height of standard cell S})$$

Therefore, any two standard cells are vertically independent.   QED

3 Constrained fuzzy graph clustering and graph bisection

Consider an edge-weighted undirected graph G(V, E) and a set of clustering constraints CC in which any pair of vertices is not grouped into the same cluster, where $V = \{x_1, x_2, \ldots, x_n\}$, $E = \{y_1, y_2, \ldots, y_m\}$ and the edge weight matrix is $C(G)$. Based on fuzzy c-means clustering, two-way constrained fuzzy graph clustering for G is proposed and applied to constrained graph bisection as follows.

3.1 Fuzzy membership and constraint

Let R^+ be the set of non-negative reals, and M_{2n} be the set of real $2 \times n$ matrices. Every function $u_i: V \to [0, 1]$ is said to assign the grade of fuzzy membership onto each x_k in V. In general, the fuzzy function u_i is called the ith fuzzy set of V. For fuzzy graph clustering, it is desired to partition V by means of fuzzy sets. Hence, two-way fuzzy graph clustering is to partition V by two fuzzy sets. Based on fuzzy c-means clustering, there will exist several constraints among the fuzzy memberships of V as follows. Given an edge-weighted graph G(V, E), two-way fuzzy graph clustering of V will be represented by a fuzzy membership matrix $U \in M_{2n}$ whose entries satisfy the following conditions:

(1) Row i of U, say $U_i = (u_{i1}, u_{i2})$, exhibits the ith fuzzy set of V.

(2) Column j of U, say $U_j = (u_{1j}, u_{2j})$, exhibits the values of the two fuzzy sets of x_j in V.

(3) u_{ik} shall be interpreted as $u_i(x_k)$, the value of the fuzzy membership of the ith fuzzy set for x_k.

(4) The sum of the values of fuzzy memberships for each x_k is 1 (column sum $\sum_i u_{ik} = 1 \; \forall k$).

(5) No fuzzy set is empty (row sum $\sum_k u_{ik} > 0 \; \forall i$).

(6) No fuzzy set is all of V (row sum $\sum_k u_{ik} < n \; \forall i$).

3.2 Clustering distance

Owing to the primary min-cut operation in two-way graph partitioning, it is desirable that any pair of vertices with larger weight will be grouped into the same cluster to reduce the partitioning cut. Hence, the larger the weight of one edge, the smaller is the clustering distance between the mapped pair of vertices. Based on this hypothesis, the clustering graph G'(V, E) is generated by modifying all the edge weights in G(V, E). For the edge $\{i, j\}$ in G, the edge weight c'_{ij} is defined as

$$c'_{ij} = \begin{cases} 1/(c_{ij} - c_{min} + 1) & \text{if } c_{min} < 1 \\ 1/c_{ij} & \text{if } c_{min} \geqslant 1 \end{cases}$$

where c_{ij} is the ijth entry of the matrix $C(G)$ and $c_{min} = \text{Min} \{c_{ij}\}$.

Furthermore, the clustering distance between any pair of unconnected vertices in G can be computed by running the shortest-path algorithm for G'. On the other hand, if there exists one clustering constraint between any pair of vertices, the clustering distance is assigned a very large number to avoid grouping the two vertices into the same cluster. Therefore, for two-way constrained fuzzy graph clustering, the clustering distance d'_{ij} between any pair of vertices, i and j, in G is further defined as

$$d'_{ij} = \begin{cases} n & \text{if the pair of vertices } i \text{ and } j \text{ is in CC} \\ c'_{ij} & \text{if } \{i, j\} \text{ is in E and the pair of vertices } i \text{ and } j \text{ is in CC} \\ \text{Short\_Path}(i, j) & \text{if } \{i, j\} \text{ is not in E and the pair of vertices } i \text{ and } j \text{ is in CC} \end{cases}$$

where $\text{Short\_Path}(s, t)$ represents the sum of weights on the shortest path from vertex s to t.

3.3 Two-way constrained fuzzy graph clustering

Based on fuzzy c-mean clustering and the definition of the clustering distance, two-way constrained fuzzy graph clustering will be transformed into the minimisation of a mathematical function $J(U, v)$. Let U be a two-way fuzzy graph clustering for G and $v = (v_1, v_2)$ be the two cluster centres, where $v_i \in V$ and $v_1 \neq v_2$. The ith clustering function $J_i: M_{2n} \times V \to R^+$ is defined as $J_i(U, v_i) = \sum_{k=1}^n (u_{ik})^2(d_{ik})^2$. Furthermore, the clustering function $J: M_{2n} \times V^2 \to R^+$ is defined as

$$J(U, v) = J_1(U, v_1) + J_2(U, v_2)$$

$$= \sum_{k=1}^n \sum_{i=1}^2 (u_{ik})^2(d_{ik})^2$$

where $U \in M_{2n}$, $v \in V^2$ with $v_i \in V$, $1 \leqslant i \leqslant 2$ and $d_{ik} = \|x_k - v_i\|$ is the clustering distance between x_k and v_i. In this function, the squared clustering distance is weighted by the 2nd power of the membership of vertex x_k in the ith cluster. Thus, the minimisation of the function $J(U, v)$ produces a fuzzy membership matrix U that is optimal in a generalised least-squared error.

Basically, from the construction of the clustering function, the function $J(U, v)$ is a nonlinear multivariable function so that it is difficult to minimise $J(U, v)$. In general, for the minimisation of such a function $J(U, v)$, a

variable-iterative optimisation approach on U and v is applied to approximate the minimum of the clustering function. Hence, the necessary conditions of U and v in the variable-iterative optimisation approach are given as follows.

Lemma 2: Consider the following problem.

Minimise $J(U, v)$,

subject to:

$$0 \leqslant u_{ik} \leqslant 1, \text{ for } 1 \leqslant k \leqslant n, 1 \leqslant i \leqslant 2$$

$$0 < \sum_{k=1}^{n} u_{ik} < n \quad \text{for } 1 \leqslant i \leqslant 2$$

$$u_{1k} + u_{2k} = 1 \quad \text{for } 1 \leqslant k \leqslant n$$

where v is fixed; then $U = \{u_{ik}\}$ is a global minimum of the problem

If $(x_k \neq v_1 \text{ AND } x_k \neq v_2)$

$$u_{ik} = \frac{d_{1k}^2 * d_{2k}^2}{d_{ik}^2(d_{1k}^2 + d_{2k}^2)} \quad \text{for } 1 \leqslant i \leqslant 2, 1 \leqslant k \leqslant n$$

Otherwise

$$u_{ik} = 1 \qquad \text{if } x_k = v_i,$$
$$= 0 \qquad \text{if } x_k \neq v_i$$
$$\text{for } 1 \leqslant i \leqslant 2, 1 \leqslant k \leqslant n$$

Lemma 3: Consider the following problem

Minimise $J(U, v)$

subject to:

$$0 \leqslant u_{ik} \leqslant 1 \quad \text{for } 1 \leqslant k \leqslant n, 1 \leqslant i \leqslant 2$$

$$0 < \sum_{k=1}^{n} u_{ik} < n \quad \text{for } 1 \leqslant i \leqslant 2$$

$$u_{1k} + u_{2k} = 1 \quad \text{for } 1 \leqslant k \leqslant n$$

where U is fixed; then $v = (v_1, v_2)$ is a global minimum of the problem if

$$v_i = x_j$$

$$\min_{x_j \in V} \left\{ \sum_{k=1}^{n} (u_{ik})^2 (d'_{jk})^2 \right\} \quad \text{for } 1 \leqslant i \leqslant 2, x_j \in V$$

As mentioned above, two-way constrained fuzzy graph clustering via the iterative optimisation of $J(U, v)$ produces a fuzzy membership matrix U for V. The basic steps of the algorithm Two-Way_Constrained_Fuzzy_Graph_Clustering (TCFGC) are given as follows.

Algorithm Two-Way_Constrained_Fuzzy_Graph_Clustering (TCFGC)
Input: an edge-weighted undirected graph $G(V, E)$ and a set of clustering constraints CC;
Begin
  (1) Determine the clustering distance d'_{ij} between x_i and x_j, $1 \leqslant i, j \leqslant n$;
  (2) Initialise an arbitrary partition and establish a fuzzy membership matrix U;
  (3) Calculate the centres $v = (v_1, v_2)$ using U as follows:
    (1) Determine v_{11} and v_{12} such that $J_1(U, v_{11})$ is the least and $J_1(U, v_{12})$ is the second least,
    (2) Determine v_{21} and v_{22} such that $J_2(U, v_{21})$ is the least and $J_2(U, v_{22})$ is the second least,

  (3) If $(v_{11} \neq v_{21})$ Then $v = (v_{11}, v_{21})$;
    Elseif $(J_1(U, v_{11}) + J_2(U, v_{22})$
$$< J_2(U, v_{21}) + J_1(U, v_{12}))$$
      Then $v = (v_{11}, v_{22})$;
      Else $v = (v_{21}, v_{12})$;
    Endif
    Endif
  (4) Calculate a new fuzzy matrix U' using $v = (v_1, v_2)$ as follows:
    If $(x_k \neq v_1 \text{ AND } x_k \neq v_2)$

$$u'_{ik} = \frac{d_{1k}^2 * d_{2k}^2}{d_{ik}^2(d_{1k}^2 + d_{2k}^2)} \quad \text{for } 1 \leqslant i \leqslant 2, 1 \leqslant k \leqslant n$$

    Otherwise

$$u'_{ik} = 1 \qquad \text{if } x_k = v_i$$
$$= 0 \qquad \text{if } x_k \neq v_i$$
$$\text{for } 1 \leqslant i \leqslant 2, 1 \leqslant k \leqslant n$$

  (5) Compare u_{ik} and u'_{ik}, for $1 \leqslant i \leqslant 2, 1 \leqslant k \leqslant n$; if $\|u'_i - u_{ik}\| < \varepsilon$, stop; otherwise, $U = U'$, and Goto step (3).
End

3.4 Constrained graph bisection

Since the algorithm TCFGC generates two groups of fuzzy memberships in U for G, the constrained graph bisection of G will be obtained by further processing the two groups of fuzzy memberships. Basically, according to the values of these fuzzy memberships in the first group, the vertices in V can be sorted decreasingly into a vertex list. Clearly, the vertex list indicates the possibility of belonging to the first cluster. In contrast, the vertex list also indicates the possibility of not belonging to the second cluster. Finally, for the constrained graph bisection, the vertex list is further separated into two vertex subsets with even size. Here, the steps of the algorithm Constrained_Graph_Bisection are described as follows.

Algorithm Constrained_Graph_Bisection
Input: A fuzzy membership matrix $U = \{u_{ik}\}$ of G;
Begin
  (1) Sort the vertex set $\{x_1, x_2, \ldots, x_n\}$ according to the fuzzy memberships, $u_{1i}, 1 \leqslant i \leqslant n$, in the first group and construct a vertex list $x_1^*, x_2^*, \ldots, x_n^*$;
  (2) Partition the vertex set $\{x_1, x_2, \ldots, x_n\}$ into $\{x_1^*, \ldots, x_{n/2}^*\}$ and $\{x_{n/2+1}^*, \ldots, x_n^*\}$, and calculate the cut of the partition.
End

4 Orientation assignment of standard cells

For the orientation assignment of standard cells, the orientation problem has been divided into the HO problem and the VO problem according to the introduction of the Manhattan net model. Furthermore, by Lemma 1, any two standard cells are vertically independent for the VO problem. Therefore, the VO problem will be solved by independently assigning vertical orientations of standard cells in $O(n)$ time, where n is the number of standard cells. Clearly, the orientation problem in a standard cell layout focuses on the HO problem. In this paper we develop the transformation between the assignment of horizontal orientations and the constrained graph bisection problem to assign horizontal orientations of standard cells.

First, by the relation of horizontal independence, all the standard cells will be divided into several horizontal dependent groups for the HO problem. It is clear that if

two cells are located on two groups, the two cells will be independently assigned horizontal orientations. Furthermore, for the assignment of horizontal orientations in one dependent group, the standard cells and the horizontal wire reduction between any pair of standard cells in this group are applied to construct an edge-weighted undirected graph. Hence, all the horizontal dependent groups are applied to construct independent edge-weighted undirected graphs.

Basically, the transformation between the ith dependent group and an edge-weighted undirected graph $G_i(V, E)$ are divided into two separate steps: vertex construction and edge construction. In the vertex construction step, the cells in the ith group are numbered as $1, 2, 3, ..., n_i$, where n_i is the number of cells in this group. As mentioned above, each standard cell S has two horizontal states, $p_s = 0$ and 1. Thus each cell S in the group will be mapped into two vertices, S(0) and S(1), in G_i. Clearly, the number of vertices in G_i is $2n_i$. On the other hand, in the edge construction step, the horizontal wire reduction between any pair of standard cells is applied to reflect the possible flipping reduction for the two cells. For any pair of vertices, $S(p_s)$ and $T(p_t)$, if $S \neq T$, one edge between $S(p_s)$, and $T(p_t)$ is constructed and the weight assigned by $FR_{hor}(S(p_s), T(p_t))$. On the other hand, if $S = T$, one clustering constraint between $S(p_s)$ and $T(p_t)$ is constructed. Since there are no two horizontal states on one cell at the same time, the clustering constraint shows that $S(p_s)$ and $T(p_t)$ are partitioned into the same cluster. Hence, for G_i, a set of clustering constraints CC_i is constructed. Clearly, the number of clustering constraints in G_i is n_i. In Fig. 9 one horizontal dependent group and the constructed graph will be illustrated.

Basically, from the construction of $G_i(V, E)$, it will take $O(n_i)$ to number the standard cells in the ith group and construct the vertex set V in the vertex construction step. Furthermore, in the edge construction step, it will take $O(n_i^2)$ time to construct the weighted edges E and a set of clustering constraints CC_i in G_i. Therefore, for a standard cell layout, it will take $\sum_i O(n_i^2)$ time to transform all independent groups in the HO problem into edge-weighted graphs. In the worst case, the time complexity of the transformation is $O(n^2)$, where n is the number of standard cells in a standard cell layout.

Based on the construction of all the independent graphs with clustering constraints, it is clear that the constrained graph bisection for G_i forms two kinds of orientation assignments for the ith independent group. Furthermore, one of the two kinds of orientation assignments is successfully decided by comparing total wire length for the ith independent group. Therefore, the HO problem in a standard cell layout is transformed into the constrained graph bisection problem for all the constructed graphs. Since the HO problem is transformed into the constrained graph bisection problem, as mentioned above, the solution of the constrained graph bisection is obtained by the application of two-way constrained fuzzy graph clustering. If the constrained graph bisection has been obtained for all constructed graphs, the HO problem in a standard cell layout is solved. Now, the algorithm Orientation_Assignment describes the solution of the orientation problem in a standard cell layout as follows.

Algorithm Orientation_Assignment
Input: A standard cell layout;
Begin
  Assign vertical orientations of standard cells;

All the standard cells are separated into several horizontal dependent groups according to the relation of horizontal dependence;
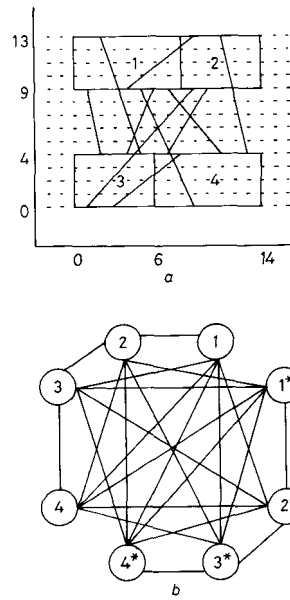

Fig. 9

a HO problem for one horizontal dependent group
  Net 1: (1, 9)–(2, 4)
  Net 2: (2, 13)–(5, 4)
  Net 3: (4, 9)–(9, 13)
  Net 4: (5, 9)–(9, 0)
  Net 5: (6, 9)–(4, 4)
  Net 6: (7, 9)–(11, 4)
  Net 7: (1, 0)–(9, 9)
  Net 8: (7, 4)–(10, 9)
  Net 9: (13, 4)–(11, 13)
  Net 10: (3, 0)–(8, 4)
b Mapped edge-weighted graph
  $W_{1, 2} = 0$, $W_{1, 3} = 0$, $W_{1, 4} = 0$
  $W_{2, 3} = 0$, $W_{2, 4} = 0$, $W_{3, 4} = 0$
  $W_{1, 2^*} = -4$, $W_{1, 3^*} = -2$, $W_{1, 4^*} = 0$
  $W_{2, 1^*} = 0$, $W_{2, 3^*} = 4$, $W_{2, 4^*} = -2$
  $W_{3, 1^*} = -2$, $W_{3, 2^*} = -4$, $W_{3, 4^*} = -2$
  $W_{4, 1^*} = -8$, $W_{4, 2^*} = -2$, $W_{4, 3^*} = 0$
  $W_{1^*, 2^*} = -4$, $W_{1^*, 3^*} = -2$, $W_{1^*, 4^*} = -8$
  $W_{2^*, 3^*} = 0$, $W_{2^*, 4^*} = 0$, $W_{3^*, 4^*} = -4$

  For (any horizontal dependent group)
    Begin
      Transform the dependent group into an edge-weighted graph with clustering constraints;
      Solve the constrained graph bisection problem for the weighted graph;
      Assign horizontal orientations of standard cells for the mapped group according to the result of the constrained graph bisection;
    End
End

5 Experimental results

The algorithm Orientation_Assignment and an optimal exhaustive algorithm have been implemented using standard C language and run on a SUN workstation under the Berkeley 4.2 UNIX operating system. Owing to the lack of benchmarks for the orientation assignment of standard cells, we create some standard cell layouts to measure the performance of the proposed algorithm.

162

Table 1 : Experimental results

Example	Cells	Initial length	Optimal		Our proposed algorithm		
			Length	Reduction	Length	Reduction	$\gamma_{optimal}$
EX1	12	183	154	15.9%	154	15.9%	100%
EX2	24	417	349	16.3%	352	15.6%	95.6%
EX3	36	725	615	15.2%	618	14.8%	97.3%
EX4	48	1073	814	24.1%	821	23.5%	97.3%
EX5	60	1394	979	29.8%	979	29.8%	100%

Furthermore, for any tested standard cell layout, we define the optimal ratio $\gamma_{optimal}$ to measure the optimal degree of the proposed algorithm to the optimal exhaust-

Fig. 10

a Initial configuration of example EX1
b Final orientation assignment
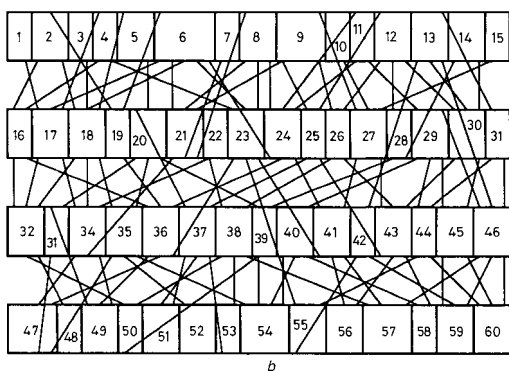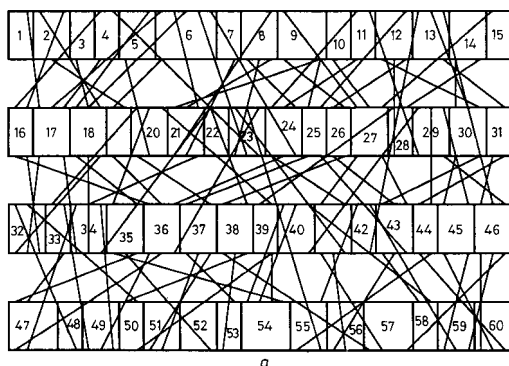
Fig. 11

a Initial configuration of example EX5
b Final orientation assignment

ive algorithm for one tested layout, such as

$$\gamma_{optimal}(Ex) = \frac{d_{near-optimal}(Ex)}{d_{optimal}(Ex)} \times 100\%$$

where $d_{optimal}(Ex)$ is the wire reduction of the layout Ex by an optimal exhaustive algorithm and $d_{near-optimal}(Ex)$ is the wire reduction of the layout Ex by the proposed algorithm.

For all the tested examples, according to the definition of the optimal ratio, all the optimal ratios are computed. From the numerical results on 50 tested examples, it is clear that the proposed algorithm has solved the orientation problem in a standard cell layout with 95% ~ 100% optimal ratio. In Table 1 the numerical results of some tested examples with the range of different sizes are computed and listed. In Fig. 10 the orientation problem for example EX1 with 12 standard cells is solved optimally by the proposed algorithm. Furthermore, example EX5 with 60 standard cells is also assigned optimal orientations of standard cells, the initial configuration and the optimal solution being shown in Fig. 11.
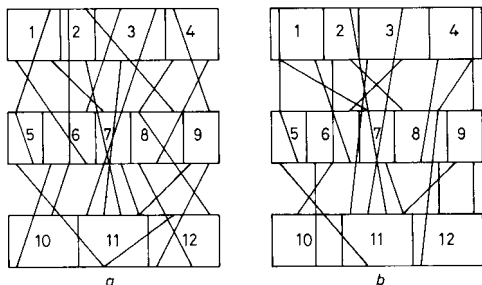
6 Conclusion

In standard cell design style, the main objective of the placement phase is to position a set of standard cells on the chip surface with minimum area or total wire length. It is well known that minimising total wire length reduces routing area in a standard cell layout. After the placement phase, another advanced improvement on total wire length is done by assigning the orientations of standard cells. In this paper, first, based on the hypothesis of the Manhattan net model, the orientation problem in a standard cell layout can be divided into the VO problem and the HO problem. Furthermore, according to the physical property in a standard cell layout, the VO problem is solved optimally in $O(n)$ time. On the other hand, based on fuzzy c-means clustering, we develop two-way constrained fuzzy graph clustering and the transformation between the orientation assignment of standard cells and the constrained graph bisection to minimise total wire length for the HO problem. Finally, the proposed algorithm has tested several standard cell layouts, and the experimental results show that the proposed algorithm produces significant wire reduction on total wire length.

7 References

1 WONG, D.F., and LIU, C.L.: 'A new algorithm for floorplan design'. 23rd *Design automation* conference, 1986, pp. 101–107
2 PREAS, B., and LORENZETTI, M. (Ed.): 'Physical design automation of VLSI systems' (Benjamin/Cummings, Menlo Park, CA, 1988)
3 PERSKY, G., DEUTSCH, D.N., and SCHWEIKERT, D.G.: 'LTX-A minicomputer-based system for automatic LSI layout'. 14th *Design automation* conference, 1977, pp. 217–255
4 YAMADA, M., and LIU, C.L.: 'An analytical method for optimal module orientation'. International symposium on *Circuits and Systems*, 1988, pp. 1679–1682

IEE Proc.-Comput. Digit. Tech., Vol. 142, No. 2, March 1995

163

5 HADAS, R.L., and LIU, C.L.: 'Solution to the module orientation and rotation problem by neural computation networks'. 26th *Design automation* conference, 1989, pp. 400–405

6 YAO, X., and LIU, C.L.: 'Solution of a module orientation and rotation problem'. European *Design automation* conference, 1990, pp. 584–588

7 CHENG, C.K., HU, T.C., and YAO, S.Z.: 'The modular orientation of VLSI layout'. International symposium on *Circuits and systems*, 1990, pp. 1600–1603

8 JEONG, J.C., and KYUNG, C.M.: 'Finding optimal module orientations in macro cell placement'. International symposium on *Circuits and Systems*, 1990, pp. 3118–3121

9 CHONG, K., and SAHNI, S.: 'Minimizing total wire length by flipping modules', *IEEE Trans.*, 1993, **CAD-12**, pp. 167–175

10 ISMAIL, M.A., and SELIM, S.Z.: 'Fuzzy C-means: optimality of solutions and effective termination of the algorithm', *Pattern Recognit.*, 1986, **19**, pp. 481–485

11 KIM, T., BEZDEK, J.C., and HATHAWAY, R.J.: 'Optimality tests for fixed points of the fuzzy C-means algorithm', *Pattern Recognit.*, 1988, **21**, pp. 651–663

12 DUNLOP, A.E.: 'A procedure for placement of standard cell VLSI circuits', *IEEE Trans.*, 1985, **CAD-4**, pp. 92–98

13 GAREY, M.R., and JOHNSON, D.S.: 'Computer and intractability: a guide to the theory of NP-completeness' (W.H. Freeman New York, NY, 1979)

14 PREAS, B.T., and KARGER, P.G.: 'Automatic placement: a review of current techniques'. 23rd *Design automation* conference, 1986, pp. 622–629

15 YAN, J.T., and HSIAO, P.Y.: 'Orientation assignment of standard cells using a fuzzy mathematical transformation'. IEEE TENCON'94, 1994, pp. 1014–1019

164

IEE Proc.-Comput. Digit. Tech., Vol. 142, No. 2, March 1995