Placement with Symmetry Constraints for Analog Layout Design Using TCG-S*

Jai-Ming Lin[1], *Guang-Ming Wu*[2], *Yao-Wen Chang*[3], *and Jen-Hui Chuang*[4]
[1]Realtek Semiconductor Corp., Science-Based Industrial Park, Hsinchu, Taiwan
[2]Department of Information Management, Nan-Hua University, Chiayi, Taiwan
[3]Department of Electrical Engineering & Graduate Institute of Electronics Engineering, National Taiwan University, Taipei 106, Taiwan
[4]Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan

Abstract

In order to handle device matching for analog circuits, some pairs of modules need to be placed symmetrically with respect to a common axis. In this paper, we deal with the module placement with symmetry constraints for analog design using the Transitive Closure Graph-Sequence (TCG-S) representation. Since the geometric relationships of modules are transparent to TCG-S and its induced operations, TCG-S has better flexibility than previous works in dealing with symmetry constraints. We first propose the necessary and sufficient conditions of TCG-S for symmetry modules. Then, we propose a polynomial-time packing algorithm for a TCG-S with symmetry constraints. Experimental results show that the TCG-S based algorithm results in the best area utilization.

1 Introduction

In the design of analog circuits, it is often required that modules (devices) be placed symmetrically with respect to one or several common axes. If the parasitics in differential analog circuits do not match, it may lead to higher offset voltages and degraded power-supply rejection ratio [5]. Placing devices symmetrically can also reduce the circuit sensitivity to thermal gradients; failure to balance thermal couplings in a differential circuit may introduce unwanted oscillations [4]. Therefore, it is desirable to develop an efficient and effective approach to place symmetry modules for analog circuit designs.

The problem of placement with symmetry modules has been extensively studied in the literature [2, 3, 4, 5, 9, 12, 14]. Most of these works used the simulated annealing algorithm in combination with floorplan representations to handle symmetry constraints. We can classify these representations into two categories: (1) the *absolute* representation and (2) the *topological* representations.

The absolute representation was proposed by Jepsen and Gellat [7]. For this representation, each module is associated with an absolute coordinate on a gridless plane. We can operate on a module by changing its coordinate directly. The KOAN/ANAGRAM II [5], PUPPY-A [12], and LAYLA [9] systems all adopted the absolute representation to handle the placement of analog modules. The main weakness of the absolute method lies in the fact that it may generate an infeasible placement with overlapped modules. Therefore, a post-processing step must be performed to eliminate this condition, implying a longer computation time and lower solution quality.

Unlike the absolute representation that operates on modules' absolute coordinates, the topological representations describes a placement by keeping the relative positions between modules. Therefore, it is often harder, but more flexible to model the symmetry constraints using the topological representations. For the topological representations, recently, several representations that can model non-slicing foorplans including sequence pairs [13], O-tree [6], and binary trees [3, 4] were used to handle the symmetry constraints. Murata et al. in [13] used two sequences of module names, namely sequence pairs, to represent the geometric relations of modules for general floorplan design. Balasa [2] then applied the sequence pairs to deal with the symmetry constraint. Guo et al. in [6] proposed the O-tree representation for a left and bottom compacted placement. Pang et al. [14] used the O-tree representation to deal with the symmetry constraints. The feasibility of the O-tree solutions can only be detected after packing; therefore, they have to explore the whole solution space to find feasible solutions with the symmetry constraints, implying a longer running time. Balasa [3] transformed an O-tree representation into a binary tree representation for non-slicing floorplans. Unlike O-tree that does not restrict the exploration space, they propose a feasibility condition for the binary tree representation with symmetry constraints, and thus only feasible solutions are searched. Chang et al. in [1] presented a *B*-tree* representation for non-slicing floorplans. Balasa et al. recently augmented the B*-tree, called the segment tree [4], to handle the symmetry constraints efficiently. Although the tree-based representations have relatively smaller solution space and a faster packing scheme than sequence pairs, they can only represent the compacted floorplan—a proper subset of the general floorplan, which implies that the optimal solution may be lost.

In this paper, we deal with the symmetry constraints using the Transitive Closure Graph-Sequence (TCG-S) representation. One major strength of TCG-S lies in the property that the geometric relationship of modules is transparent to TCG-S and its induced operations, implying that any violation of the symmetry constraints during perturbation can easily be detected and thus infeasible solutions can be discarded earlier. Therefore, TCG-S has better flexibility than previous works in dealing with symmetry constraints. We first develop the necessary and sufficient conditions of TCG-S for symmetry modules. Then, we propose an $O(m^2)$-time packing algorithm for a TCG-S with symmetry constraints, where m is the number of modules. Experimental results show that the TCG-S based algorithm results in the best area utilization.

It should be noted that the packing times for the previous works with symmetry constraints are $O(m^2)$ for sequence pair [2], $O(m^2)$ for the O-tree [14], and $O(m \lg m)$ for the B*-tree (the segment tree) [4]. The B*-tree based method has lower packing time complexity, but, like the O-tree, it is not P*-admissible [11] and can handle only compacted floorplan (which is a proper subset of the general floorplan).

The remainder of this paper is organized as follows. Section 2 formulates the placement problem with symmetry constraints. Section 3 reviews the TCG-S representation. Section 4 presents the feasibility conditions of TCG-S and a packing algorithm for placement with symmetry constraints. Section 5 introduces the perturbations for symmetry constraints. Experimental results are reported in Section 6, and concluding remarks are given in Section 7.

2 Preliminaries

Let $B = \{b_1, b_2, ..., b_m\}$ be a set of m rectangular modules whose width, height, and area are denoted by W_i, H_i, and A_i, $1 \le i \le m$. Let (X_i, Y_i) ((X'_i, Y'_i)) denote the coordinate of the bottom-left (top-right) corner of module b_i, $1 \le i \le m$, on a chip. Symmetry constraints can be formulated in terms of *symmetry pairs* and *symmetry groups*. A *symmetry pair* (b_{i_B}, b_{i_T}) is a pair of modules with the same dimensions that have to be placed symmetrically with respect to an axis, where b_{i_B} (b_{i_T}) denotes the symmetry module in the bottom (top) side. A *symmetry group* $\{(b_{1_B}, b_{1_T}), (b_{2_B}, b_{2_T}), \ldots, (b_{k_B}, b_{k_T})\}$ consists of a set of symmetry pairs. For easier presentation, we consider the common axes horizontal of symmetry groups; however, similar method can be applied if the axes are in the x axis. Let Y_{i_M} denote the middle axis for a symmetry pair (b_{i_B}, b_{i_T}) (i.e., $2Y_{i_M} = Y'_{i_B} + Y_{i_T}$). Given a symmetry group, the x coordinates of the modules in each symmetry pair must be the same. Besides, they have to be placed symmetrically with respect to a common axis in the final placement, which implies the following equations:

$$X_{i_B} = X_{i_T} \quad \forall i = 1, \ldots, k, \tag{1}$$

$$Y_g = Y_{i_M} \quad \forall i = 1, \ldots, k, \tag{2}$$

where Y_g is the coordinate of the common axis for the symmetry group. A placement P is an assignment of (X_i, Y_i), $i = 1, \ldots, m$, for each b_i such that no two modules overlap and the symmetry constraints are satisfied as well. The goal of placement with symmetry modules is to optimize a predefined cost metric such as the resulting area (i.e., the minimum bounding rectangle of P) induced by a placement.

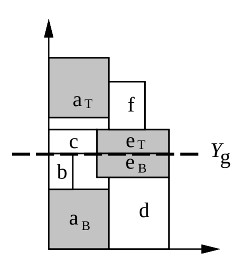

Figure 1: A feasible placement with a symmetry group $\{(b_{a_B}, b_{a_T}), (b_{e_B}, b_{e_T})\}$.

Figure 1 shows a feasible placement with symmetry modules. There are eight modules b_{a_B}, b_{a_T}, b_b, b_c, b_d, b_{e_B}, b_{e_T}, and b_f in the placement, where (b_{a_B}, b_{a_T}) and (b_{e_B}, b_{e_T}) are symmetry pairs in a symmetry group.

*Yao-Wen Chang's work was partially supported by the National Science Council of Taiwan under Grant No's. NSC 92-2215-E-002-018, NSC-93-2220-E-002-001, and NSC 93-2752-E-002-008-PAE and by a grant from SprintSoft Inc.

3 Review of TCG-S

We first review the TCG-S representation presented in [11]. TCG-S describes the geometric relations among modules based on two graphs, namely a *horizontal transitive closure graph* C_h and a *vertical transitive closure graph* C_v, and a packing sequence Γ_-. A node n_i in C_h (C_v) represents a module b_i and an edge (n_i, n_j) denotes that module b_i is left to (below) module b_j. TCG-S has the following *feasibility properties*:

1. C_h and C_v are acyclic.
2. Each pair of nodes must be connected by exactly one edge either in C_h or in C_v.
3. The transitive closure of C_h (C_v) is equal to C_h (C_v) itself.[1]
4. The packing sequence Γ_- is the topological ordering of C_h and C_v.

Figure 2(a) shows a placement with eight modules a_B, a_T, b, c, d, e_B, e_T, and f whose widths and heights are (5, 5), (5, 5), (2, 3), (4, 2), (5, 6), (6, 2), (6, 2), and (3, 4), respectively. Figure 2(b) shows the TCG-S $= (C_h, C_v, \Gamma_-)$ corresponding to the placement of Figure 2(a). The value associated with a node in C_h (C_v) gives the width (height) of the corresponding module, and the edge (n_i, n_j) in C_h (C_v) denotes the horizontal (vertical) relation of b_i and b_j. Since there exists an edge (n_{a_B}, n_d) in C_h, module a_B is left to module d. Similarly, module a_B is below module b since there exists an edge (n_{a_B}, n_b) in C_v. The packing sequence $\Gamma_- = \{a_B, b, d, e_B, c, e_T, a_T, f\}$ corresponds to the topological ordering of C_h and C_v.

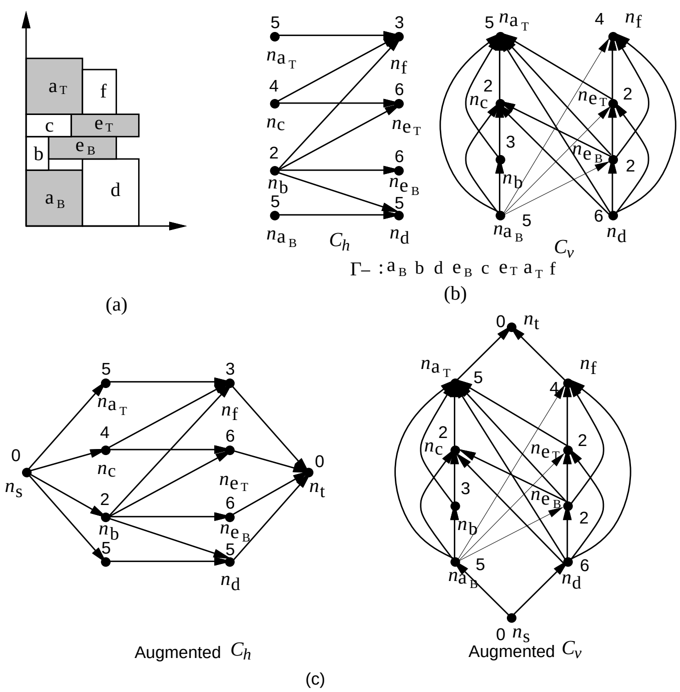

Figure 2: (a) A placement. (b) TCG-S. (c) Augmented C_h and C_v.

Given a TCG-S, a placement can be obtained in $O(m^2)$ time by performing a well-known *longest path algorithm* [10] on the TCG-S, where m is the number of modules. To facilitate the implementation of the longest path algorithm, the two closure graphs can be augmented as follows. For each closure graph, we introduce two special nodes, the source n_s and the sink n_t, both with zero weights, and construct edges from n_s to each node with in-degree equal to zero as well as from each node with out-degree equal to zero to n_t. (Note that the augmentation is performed only for packing.) Figure 2(c) shows the augmented C_h and C_v for the C_h and C_v shown in Figure 2(b). The longest path algorithm executes as follows: we process modules according to their topological ordering in the augmented C_h (C_v). The x (y) coordinate of dummy module b_s is zero (i.e., $X_s = 0$ ($Y_s = 0$)). For each module b_i in the topological ordering, we serially *relax* it as follows: $X_j = \max\{X_j, X_i + W_i\}$ ($Y_j = \max\{Y_j, Y_i + H_i\}$) if $n_j \in F_{out}(n_i)$. Let $L_h(n_i)$ ($L_v(n_i)$) denote the weight of the longest path from n_s to n_i in the augmented C_h (C_v). The coordinate (X_i, Y_i) of a module b_i is given by ($L_h(n_i)$, $L_v(n_i)$). Since the respective width and height of the placement for the given TCG-S are $L_h(n_t)$ and $L_v(n_t)$, the area of the placement is given by $L_h(n_t)L_v(n_t)$. Since each module has a unique coordinate after packing, there exists a unique placement corresponding to any TCG-S.

4 TCG-S for Symmetry Constraints

In this section, we first introduce necessary and sufficient conditions of feasible TCG-S for the symmetry constraints.

[1] The transitive closure of a directed acyclic graph G is defined as the graph $G' = (V, E')$, where $E' = \{(n_i, n_j)$: there is a path from node n_i to node n_j in $G\}$.

4.1 Feasible TCG-S

In [11], we had shown that there always exists a unique feasible placement corresponding to a TCG-S for rectangular modules. However, for a TCG-S with symmetry modules, we must satisfy Equations (1) and (2) mentioned in Section 2 in the final placement. Therefore, we shall add two additional feasibility constraints for a TCG-S with symmetry constraints as follows:

5. *Essence constraint*: for modules b_{i_B} and b_{i_T} in each symmetry pair, the corresponding edge (n_{i_B}, n_{i_T}) must be in C_v.
6. *Homology constraint*: for two arbitrary symmetry pairs (b_{i_B}, b_{i_T}) and (b_{j_B}, b_{j_T}) in a symmetry group, one of the following conditions is satisfied: (1) edges (n_{i_B}, n_{j_B}) and (n_{i_T}, n_{j_T}) are both in C_h, (2) edges (n_{j_B}, n_{i_B}) and (n_{j_T}, n_{i_T}) are both in C_h, (3) edges (n_{i_B}, n_{j_B}) and (n_{j_T}, n_{i_T}) are both in C_v, and (4) edges (n_{j_B}, n_{i_B}) and (n_{i_T}, n_{j_T}) are both in C_v.

A TCG-S is called *symmetric feasible* iff Conditions 1–6 are satisfied. For the essence constraint, if b_{i_B} and b_{i_T} are the modules in a symmetry pair, b_{i_B} must be directly below b_{i_T} in order to have the same x coordinates. There must exist an edge (n_{i_B}, n_{i_T}) in C_v according to the definition of TCG-S. In contrast, if edge (n_{i_B}, n_{i_T}) (or (n_{i_T}, n_{i_B})) is in C_h, the x coordinates of b_{i_B} and b_{i_T} cannot be the same since b_{i_B} (b_{i_T}) is in the left side of b_{i_T} (b_{i_B}). For the homology constraint, if b_{i_B} is left to b_{j_B}, b_{i_T} must be left to b_{j_T} such that the x coordinates of b_{i_B} and b_{i_T} (b_{j_B} and b_{j_T}) are the same. This implies that the edges (n_{i_B}, n_{j_B}) and (n_{i_T}, n_{j_T}) are in C_h. The x coordinates of b_{i_B} and b_{i_T} (b_{j_B} and b_{j_T}) are not the same if b_{i_B} is left to b_{j_B} but b_{i_T} is right to b_{j_T}. Also, if b_{i_T} is above b_{j_T}, b_{j_B} must be above b_{i_B} such that the middle axes of the two symmetry pairs can be between b_{j_B} and b_{j_T}. This implies that edges (n_{j_T}, n_{i_T}) and (n_{i_B}, n_{j_B}) are in C_v. In contrast, the middle axis of (b_{i_B}, b_{i_T}) is always above that of (b_{j_B}, b_{j_T}) if b_{i_T} is above b_{j_T} and b_{i_B} is above b_{j_B}.

Figures 3(a) and (b) show a TCG-S with the symmetry group $\{(b_{a_B}, b_{a_T}), (b_{e_B}, b_{e_T})\}$ and the corresponding placement. The TCG-S shown in Figure 3(a) is feasible for the symmetry modules since the essence constraint (i.e., edges (n_{a_B}, n_{a_T}) and (n_{e_B}, n_{e_T}) are in C_v) and the homology constraint (i.e., edges (n_{a_B}, n_{e_B}) and (n_{e_T}, n_{a_T}) are in C_v) are satisfied. (Note that the modules b_{e_B} and b_{e_T} shown in Figure 3(b) are not symmetric. We will discuss how to make them symmetric later.) For the TCG-S shown in Figures 3(c), Condition (1) of the homology constraint is violated since edge (n_{a_T}, n_{e_T}) is in C_h but (n_{a_B}, n_{e_B}) is not. This makes b_{a_T} in the left side of b_{e_T} and b_{a_B} in the right side of b_{e_B} in the resulting placement shown in Figures 3(d), and thus b_{a_B} and b_{a_T} (b_{e_B} and b_{e_T}) cannot have the same x coordinate. Similarly, Condition (3) of the homology constraint is violated for the TCG-S shown in Figure 3(e) because edge (n_{a_T}, n_{e_T}) is in C_v while edge (n_{e_B}, n_{a_B}) is not. Since the middle axis of (b_{e_B}, b_{e_T}) is always higher than that of (b_{a_B}, b_{a_T}) in the resulting placement shown in Figure 3(f), the two symmetry pairs cannot have the same middle axis.

4.2 Packing

As mentioned in Section 3, the packing of regular modules can be obtained by applying the longest path algorithm on the augmented transitive closure graph. To guarantee a feasible placement with symmetry constraints, however, we need to modify the packing algorithm. Figure 4 illustrates the difference between the packings with and without symmetry modules. Figures 4(a) and (b) give a TCG-S with a symmetry group $\{(b_{a_B}, b_{a_T}), (b_{e_B}, b_{e_T})\}$ and its corresponding placement. The placement is not correct for the symmetry modules because $X_{e_B} \neq X_{e_T}$ and $Y_{e_M} \neq Y_{a_M}$. The correct one is shown in Figure 4(d).

For each symmetry pair (b_{i_B}, b_{i_T}), the x coordinates of b_{i_B} and b_{i_T} must be the same. To make $X_{i_B} = X_{i_T}$, in addition to the longest path from the source to b_{i_B} (b_{i_T}) in the augmented C_h, we also have to consider the x coordinate of b_{i_T} (b_{i_B}). For the placement shown in Figure 4(b), the x coordinates of b_{e_B} and b_{e_T} are different since b_{e_T} are in the right side of both b_b and b_c while b_{e_B} is in the right side of only b_b according to the TCG-S of Figure 4(a). If we make b_{e_B} and b_{e_T} both in the right side of b_b and b_c, their x coordinates will be the same (see Figure 4(c) for the resulting placement). Let the *fan-in* (*fan-out*) of a node n_i, denoted by $F_{in}(n_i)$ ($F_{out}(n_i)$), be the nodes n_j's with edges (n_j, n_i) ((n_i, n_j)). For the modules in each symmetry pair (b_{i_B}, b_{i_T}), we have to add dummy edge (n_j, n_{i_B}) ((n_j, n_{i_T})) to C_h if $n_j \in F_{in}(n_{i_T})$ and $n_j \notin F_{in}(n_{i_B})$ ($n_j \in F_{in}(n_{i_B})$ and $n_j \notin F_{in}(n_{i_T})$) before applying the longest path algorithm on the graph. After the x coordinate of each module is determined, those newly added edges are removed from C_h to guarantee the correctness of TCG-S during perturbation.

For each symmetry pair (b_{i_B}, b_{i_T}) in a symmetry group, the distance between b_{i_B} and Y_g must equal to that between b_{i_T} and Y_g, where Y_g is the common axis of the group. Let $\Delta(i_T)$ ($\Delta(i_B)$) denote the distance between the bottom (top) boundary of module b_{i_T} (b_{i_B}) and Y_g (i.e $\Delta(i_T)= Y_{i_T} - Y_g$ ($\Delta(i_B)= Y_g - Y'_{i_B}$)). For the placement shown in Figure 4(c), two symmetry pairs (b_{a_B}, b_{a_T}) and (b_{e_B}, b_{e_T}) are not placed symmetrically with respect to the same axis since the original packing algorithm relaxes modules according to the sequence defined in Γ_- without considering $\Delta(a_B)$ and $\Delta(a_T)$ ($\Delta(e_B)$ and $\Delta(e_T)$) as their values. (See the last photography of Section 3 for the relax process.) If we first relax the symmetry pair (b_{e_B}, b_{e_T}), and then the other (b_{a_B}, b_{a_T}), we can make $\Delta(e_B)$ ($\Delta(a_B)$) equal to $\Delta(e_T)$ ($\Delta(a_T)$) by choosing the larger one of the two values as their values. We first assume that the common axis of a symmetry group is the highest middle

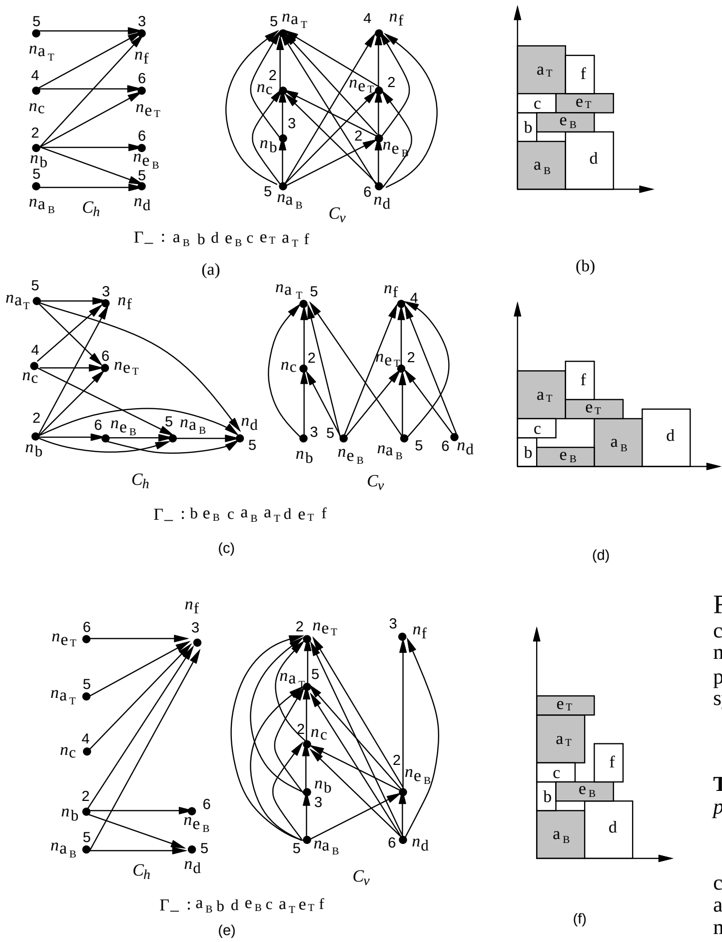

Figure 3: (a) A symmetric feasible TCG-S, (b) the placement corresponding to the TCG-S in (a), (c) a symmetric infeasible TCG-S (violation of Condition (1) of the homology constraint), (d) the placement corresponding to the TCG-S in (c), (e) a symmetric infeasible TCG-S (violation of Condition (3) of the homology constraint), (f) the placement corresponding to the TCG-S in (e).

axis Y_{j_M} for all symmetry pairs in the group. Therefore, for each symmetry pair (b_{i_B}, b_{i_T}) in a symmetry group, the initial value for the distance between bottom (top) symmetry module and the common axis is $Y_{j_M} - Y'_{i_B}$ (i.e., $\Delta(i_B) = \Delta(i_T) = Y_{j_M} - Y'_{i_B}$). Also, the distance between other module b_w and the common axis is $Y_w - Y_{j_M}$ ($Y_{j_M} - Y'_w$) if b_w is above (below) the common axis. Different from the original longest path algorithm that relaxes modules according to the sequence defined in Γ_-, we first relax modules b_{l_B} and b_{l_T} if b_{l_B} is the last bottom symmetry module in Γ_- since there cannot exist any symmetry modules between b_{l_B} and b_{l_T}. Note that a module is deleted from Γ_- if it is relaxed in this algorithm. We then relax the modules from b_{l_B} to b_{k_B} and from b_{k_B} to b_{k_T} if b_{k_B} is the nearest bottom symmetry module in the left side of b_{l_B} in Γ_-. The process is repeated until all modules in Γ_- are traversed. A module is said to be *above-marked (below-marked)* if it is a fan-out (fan-in) module of a relaxed module. During traversing modules from b_{l_B} to b_{k_B}, those below-marked modules b_u's are relaxed for those modules belonging to fan-in's of b_u's. Then, we start to relax the above-marked modules b_v's for those modules belonging to fan-out's of b_v's until the top symmetry module b_{k_T} is met. (Note that we only relax the marked modules since only those modules belonging to fan-in's (fan-out's) of symmetry modules are changed during our algorithm.) Then, we can make $\Delta(k_B)$ equal to $\Delta(k_T)$ by assigning $\Delta(k_B) = \Delta(k_T) = \max\{\Delta(k_B), \Delta(k_T)\}$. After $\Delta(k_B)$ and $\Delta(k_T)$ are obtained, b_{k_B} and b_{k_T} are relaxed and the same process repeats until the last module in Γ_- is met. After these processes, $\Delta(i_B) = \Delta(i_T)$ for each symmetry pair (b_{i_B}, b_{i_T}) in the group. If the y coordinate of the common axis is zero, the coordinates for those below-marked (above-marked) or bottom (top) symmetry modules b_p's (b_q's) are $-\Delta(p)$ ($\Delta(q)$). To make the coordinates of all modules larger than zero, we have to pull all modules upward by $\Delta(s)$ distance, and thus $Y'_p = -\Delta(p) + \Delta(s)$ ($Y_q = \Delta(q) + \Delta(s)$), where $\Delta(s)$ is distance between the dummy module b_s and the common axis. Finally, for those unprocessed modules b_x's, their new coordinates $Y_x = \max\{Y'_r\}$ if $n_r \in F_{in}(n_x)$ in C_v.

Theorem 1 *Given a symmetric feasible TCG-S, the packing scheme gives a feasible placement of the minimum area in $O(m^2)$ time, where m is the number of modules.*

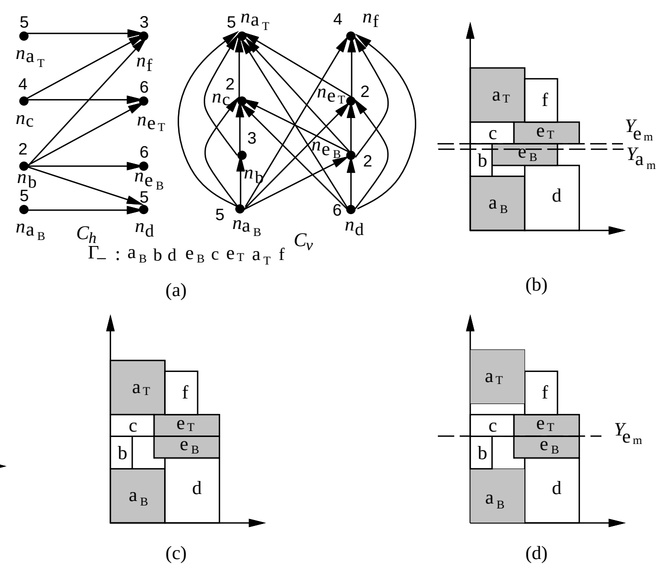

Figure 4: (a) and (b) A TCG-S and its corresponding placement without considering the symmetry pair $\{(b_{e_B}, b_{e_T})\}$. (c) The resulting placement after modules b_{e_B} and b_{e_T} are aligned at the same x coordinate. (d) The resulting placement after two symmetry pairs (b_{a_B}, b_{a_T}) and (b_{e_B}, b_{e_T}) are placed symmetrically with respect to the same axis Y_{e_m}.

Theorem 2 *There exists a correspondence between a minimum area rectangle packing with symmetry constraints and a symmetric feasible TCG-S.*

Note that the packing times for the previous works with symmetry constraints are $O(m^2)$ for sequence pair [2], $O(m^2)$ for the O-tree [14], and $O(m \lg m)$ for the B*-tree (the segment tree) [4]. The B*-tree based method has lower packing time complexity, but, like the O-tree, it is not P*-admissible [11] and can handle only compacted floorplan (which is a proper subset of the general floorplan).

5 Algorithm

Our algorithm is based on simulated annealing [8]. Given an initial solution represented by a TCG-S, we perturb the TCG-S to obtain a new TCG-S. The perturbation continues to search for a "good" configuration until a predefined termination condition is satisfied. To ensure the correctness of a packing with symmetry modules, the new TCG-S must satisfy the TCG-S feasibility conditions described in Section 3 and the essence and the homology constraints presented in Section 4.1. The following five operations are introduced in [11] to perturb a TCG-S:

- *Rotation:* Rotate a rectangular module.
- *Swap:* Swap two nodes associated with two rectangular modules in both C_h *and* C_v.
- *Reverse:* Reverse a *reduction edge* in C_h *or* C_v.
- *Move:* Move a *reduction edge* from one transitive closure graph (C_h or C_v) to the other.
- *Transpositional Move:* Move a *reduction edge* from one transitive closure graph (C_h or C_v) to the other, and then transpose the two nodes associated with the edge.

We introduce the five operations that can perturb an arbitrary edge, and we show how to maintain the feasibility of symmetric feasible TCG-S in each perturbation as follows.

5.1 Rotation

We cannot rotate node n_i if b_i denotes a symmetry module except the other symmetry module b_j is rotated.

5.2 Swap

We cannot swap two nodes n_i and n_j if b_i or b_j denotes a symmetry module except the following two conditions:

- n_i and n_j denote modules in a symmetry pair.
- n_i and n_j denote two bottom (top) symmetry modules. Then, we have to swap the corresponding top (bottom) symmetry modules.

For the first condition, b_i (b_j) is considered as top (bottom) symmetry module if it is a bottom (top) symmetry module originally.

5.3 Reverse

If we reverse a reduction edge (n_i, n_j) (there does not exist another path from n_i to n_j, except the edge (n_i, n_j) itself) in a transitive closure graph G, the feasibility of TCG-S is violated under the following two conditions:

- $G = C_h$**:** one fan-in of n_j or n_j denotes a bottom (top) symmetry module b_{p_B} (b_{p_T}), and one fan-out of n_i or n_i denotes the other top (bottom) symmetry module b_{p_T} (b_{p_B}).

Circuit	# of mod.	symmetry modules	Total area	Sequence pairs		Segment tree		TCG-S	
				Area	Time (sec)	Area	Time (sec)	Area (%dead space)	Time (sec)
apte	9	8	46.56	48.12	25	47.52	11	47.52 (2.1%)	3
hp	11	8	8.830	9.84	138	9.714	62	9.714 (10.0%)	50
ami33	33	6	1.156	1.24	684	1.23	307	1.21 (4.7%)	423
ami49	49	4	35.45	37.82	2038	37.31	983	37.04 (4.5%)	1247

Table 1: Experimental results for the benchmark circuits.

- $G = C_v$**:** one fan-in of n_j or n_j denotes a top symmetry module b_{p_T}, and one fan-out of n_i or n_i denotes another bottom symmetry module b_{q_B}.

If we reverse a reduction edge (n_{i_B}, n_{j_B}) (or (n_{i_T}, n_{j_T})) in C_h, the edge (n_{i_T}, n_{j_T}) ((n_{i_B}, n_{j_B})) must also be reversed to maintain the homology constraint. Besides, if a new edge (n_{l_B}, n_{k_B}) (or (n_{l_T}, n_{k_T})) is added to a C_h during the operation, we must also move the edge (n_{l_T}, n_{k_T}) ((n_{l_B}, n_{k_B})) from C_v to C_h if there exists an edge (n_{l_T}, n_{k_T}) ((n_{l_B}, n_{k_B})) is in C_v; otherwise, we should transpositionally move edge (n_{k_T}, n_{l_T}) ((n_{k_B}, n_{l_B})) from C_v to C_h if the edge is in C_v. Similarly, if we reverse a reduction edge (n_{i_B}, n_{j_B}) (or (n_{i_T}, n_{j_T})) in C_V, the edge (n_{j_T}, n_{i_T}) ((n_{j_B}, n_{i_B})) must also be reversed. Besides, if a new edge (n_{l_B}, n_{k_B}) ((n_{l_T}, n_{k_T})) is added to a C_v during this operation, we must also move the edge (n_{k_T}, n_{l_T}) ((n_{k_B}, n_{l_B})) to C_v if there exists an edge (n_{k_T}, n_{l_T}) ((n_{k_B}, n_{l_B})) in C_h; otherwise, we should transpositionally move an edge (n_{l_T}, n_{k_T}) ((n_{l_B}, n_{k_B})) from C_h to C_v if the edge is in C_h.

5.4 Move

If we move a reduction edge (n_i, n_j) from a transitive closure graph G to the other G', the feasibility of TCG-S with symmetry modules is violated in the following conditions:

- $G' = C_h$**:** one fan-in of n_i or n_i denotes a bottom (top) symmetry module b_{p_B} (b_{p_T}), and one fan-out of n_j or n_j denotes the other top (bottom) symmetry module b_{p_T} (b_{p_B}).
- $G = C_v$**:** one fan-in of n_i or n_i denotes a top symmetry module b_{p_T}, and one fan-out of n_j or n_j denotes another bottom symmetry module b_{q_B}.

It takes $O(m)$ time to detect the violation by checking the modules b_q's before b_i and after b_j according the sequence defined in Γ_-, where m is number of modules.

If we move edge a reduction edge (n_{i_B}, n_{j_B}) (or (n_{i_T}, n_{j_T})) from C_v to C_h, the edge (n_{j_T}, n_{i_T}) ((n_{j_B}, n_{i_B})) in C_v must be transpositionally moved to C_h to maintain the homology constraint. Besides, if a new edge (n_{l_B}, n_{k_B}) (or (n_{l_T}, n_{k_T})) is added to a C_h during the operation, we must move the edge (n_{l_T}, n_{k_T}) ((n_{l_B}, n_{k_B})) to C_h if there exists an edge (n_{l_T}, n_{k_T}) ((n_{l_B}, n_{k_B})) in C_v; otherwise, we should transpositionally move edge (n_{k_T}, n_{l_T}) ((n_{k_B}, n_{l_B})) from C_v to C_h if the edge is in C_v. Similarly, if we move a reduction edge (n_{i_B}, n_{j_B}) (or (n_{i_T}, n_{j_T})) from C_h to C_v, the edge (n_{i_T}, n_{j_T}) ((n_{i_B}, n_{j_B})) in C_h must be transpositionally moved to C_v. Besides, if a new edge (n_{l_B}, n_{k_B}) ((n_{l_T}, n_{k_T})) is added to a C_v during this operation, we must move the edge (n_{k_T}, n_{l_T}) ((n_{k_B}, n_{l_B})) to C_v if there exists an edge (n_{k_T}, n_{l_T}) ((n_{k_B}, n_{l_B})) in C_h; otherwise, we should transpositionally move edge (n_{l_T}, n_{k_T}) ((n_{l_B}, n_{k_B})) from C_h to C_v if the edge is in C_h.

5.5 Transpositional Move

If we transpositionally move a reduction edge (n_i, n_j) from a transitive closure graph G to the other G', the feasibility of TCG-S with symmetry modules is violated in the following conditions:

- $G' = C_h$**:** one fan-in of n_j or n_j denotes a bottom (top) symmetry module b_{p_B} (b_{p_T}), and one fan-out of n_i or n_i denotes the other top (bottom) symmetry module b_{p_T} (b_{p_B}).
- $G' = C_v$**:** one fan-in of n_j or n_j denotes a top symmetry module b_{p_T}, and one fan-out of n_i or n_i denotes another bottom symmetry module b_{q_B}.

If we transpositionally move a reduction edge (n_{i_B}, n_{j_B}) (or (n_{i_T}, n_{j_T})) from C_v to C_h, the edge (n_{j_T}, n_{i_T}) ((n_{j_B}, n_{i_B})) in C_v must be also moved to C_h. Besides, if a new edge (n_{l_B}, n_{k_B}) (or (n_{l_T}, n_{k_T})) is added to a C_h during the operation, we must move the edge (n_{l_T}, n_{k_T}) ((n_{l_B}, n_{k_B})) to C_h if there exists an edge is (n_{l_T}, n_{k_T}) ((n_{l_B}, n_{k_B})) in C_v; otherwise, we should transpositionally move edge (n_{k_T}, n_{l_T}) ((n_{k_B}, n_{l_B})) to C_h if the edge is in C_v. Similarly, if we transpositionally move a reduction edge (n_{i_B}, n_{j_B}) (or (n_{i_T}, n_{j_T})) from C_h to C_v, the edge (n_{i_T}, n_{j_T}) ((n_{i_B}, n_{j_B})) in C_h is also moved to C_v. Besides, if a new edge (n_{l_B}, n_{k_B}) ((n_{l_T}, n_{k_T})) is added to a C_v during this operation, we must move the edge (n_{k_T}, n_{l_T}) ((n_{k_B}, n_{l_B})) to C_v if there exists an edge is (n_{k_T}, n_{l_T}) ((n_{k_B}, n_{l_B})) in C_h; otherwise, we should transpositionally move an edge (n_{l_T}, n_{k_T}) ((n_{l_B}, n_{k_B})) to C_v if the edge is in C_v.

6 Experimental Results

Based on the simulated annealing method [8], we implemented the placement algorithm using TCG-S in the C++ programming language on a 433 MHz SUN Sparc Ultra-60 workstation with 1 GB memory. The benchmarks (apte, hp, ami33, and ami49) in Table 1, we impose the symmetry constraints to the modules with the same dimensions in a set of commonly used MCNC benchmarks. As shown in Table 1, Columns 2 lists the number of modules of four benchmarks. Columns 3 lists the number of modules in symmetric group.

The total area of modules in each circuit are shown in Column 4. Columns 5, 7, and 9 list the respective resulting areas obtained by the sequence pairs (SP) method proposed in [2], the segment-tree method presented in [4], and our program, *based on the same simulated annealing engine*; Columns 6, 8, and 10 list their respective running times. As shown in the table, our method resulted in much more effective area utilization than the SP-based and the segment tree-based methods. Further, our method is also much more efficient than the SP-based method and comparable to the segment tree-based method. Figure 5 shows the resulting placement for ami33 (the shaded modules denote symmetry modules). The experimental results show that our TCG-based algorithm consistently obtained good results.

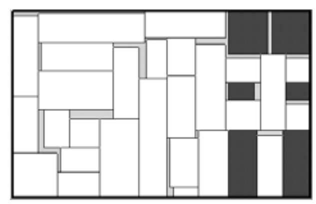

Figure 5: The resulting placement of ami33 with three symmetry pairs (area $= 1.219mm^2$).

7 Concluding Remarks

We have presented a TCG-S based algorithm to deal with the placement with symmetry constraints. TCG-S is the first *general* graph representation with the feasibility guarantee for each perturbation. We have derived necessary and sufficient conditions of TCG-S for symmetry modules, and proposed a packing algorithm for TCG-S. Experimental results have shown that our method is very efficient and effective.

References

[1] Y.-C. Chang, et al, “B*-trees: A New Representation for Non-Slicing Floorplans,” *Proc. DAC*, pp. 458–463, 2000.

[2] F. Balasa and K. Lampaert, ”Symmetry Within the Sequence-Pair Representation in the Context of Placement for Analog Design,” *IEEE TCAD*, vol. 19, no. 7, 721–731, July 2000.

[3] F. Balasa, ”Modeling Non-Slicing Floorplans with Binary Trees,” *ICCAD*, 2000.

[4] F. Balasa, et al, ”Efficient solution space exploration based on segment trees in analog placement for analog placement with sysmetry constraints,” *ICCAD*, pp. 497–502, 2002.

[5] J. Cohn, et al, “KOAN/ANAGRAMII: New Tools for Device-Level Analog Layout,” *IEEE J. Solid-State Circuits*, vol 26., PP. 330-342, Mar. 1991.

[6] P.-N. Guo, et al, “An O-Tree Representation of Non-Slicing Floorplan and Its Applications,”*Proc. DAC*, pp. 268–273, 1999.

[7] D. W. Jepsen and C. D. Gellat Jr., “Macro Placement by Monte Carlo Annealing,” *Proc. ICCD*, pp. 495-485, Nov. 1983.

[8] S. Kirkpatrick, et al, “Optimization by Simulated Annealing,” *Science*, vol. 220, no. 4598, May 13, 1983, pp.671–680.

[9] K. Lampaert, et al, “A Performance-Driven Placement Tool for Analog Integrated Circuits,” *IEEE J. Solid-State Circuits*, vol. 30, pp. 773-780, July 1995.

[10] E. Lawler, *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart, and Winston, 1976.

[11] J.-M. Lin and Y.-W. Chang, “TCG-S: Orthogonal Coupling of P*-admissible Representations for General Floorplans,” *IEEE Trans. Computer-Aided Design*, Vol. 24, No. 6, June 2004.

[12] E. Malavasi, et al, “Automation of IC Layout with Analog Constraints,” *IEEE TCAD*, vol. 15, pp. 923-942, Aug. 1996.

[13] H. Murata, et al, “Rectangle-Packing Based Module Placement,” *Proc. ICCAD*, pp. 472–479, 1995.

[14] Y. Pang, et al, “Block Placement with Symmetry Constraints based on thee O-tree Non-Slicing Representation,” *Proc. DAC*, pp. 464-467, 2000.