Advances in Radio Science (2003) 1: 191–196

Advances in Radio Science

A novel bottom-left packing genetic algorithm for analog module placement

L. Zhang and U. Kleine

Otto-von-Guericke University of Magdeburg, IESK, PO Box 4120, D-39016 Magdeburg, Germany

Abstract. This paper presents a novel genetic algorithm for analog module placement. It is based on a generalization of the two-dimensional bin packing problem. The genetic encoding and operators assures that all constraints of the problem are always satisfied. Thus the potential problems of adding penalty terms to the cost function are eliminated, so that the search configuration space decreases drastically. The dedicated cost function covers the special requirements of analog integrated circuits. A fractional factorial experiment was conducted using an orthogonal array to study the algorithm parameters. A meta-GA was applied to determine the optimal parameter values. The algorithm has been tested with several local benchmark circuits. The experimental results show this promising algorithm makes the better performance than simulated annealing approach with the satisfactory results comparable to manual placement.

1 Introduction

The significant tendency of system-on-chip (SoC) intensifies the booming market share of mixed-signal integrated circuits (ICs). Although most of the functions in such an integrated system are performed with digital circuitry, analog circuits are always needed as an interface to the external, continuous-valued world. The design of digital portion can be tackled with modern cell-based tools (Wang et al., 2000) for synthesis, mapping and physical design. The analog counterpart, however, is still routinely designed by hand. The layout of analog circuits is intrinsically more difficult than the digital one. To address this problem, an automated layout tool called ALADIN (Automatic Layout Design Aid for Analog Integrated Circuits) (Zhang et al., 2000) is currently being developed for analog experts who can and must bring their specific knowledge into the synthesis process in order to create high quality layouts. The focus of this paper is on the placement phase. Due to analog constraints such as matching requirements, it is usually preferred to build more or less complex clusters of devices, hereafter called modules or macro-cells, which are parameterized for the processed sub-circuits. The objective of placement is to position the modules appropriately so that the chip area and the total wire length of the interconnections are minimized under certain constraints.

Many heuristic strategies for module placement based on iterative improvement have been published so far, such as force-directed, min-cut, passive resistive optimization, simulated annealing (SA) (Su et al., 2001; Plas et al., 2001) and genetic algorithm (GA) (Shahookar and Mazumder, 1990; Esbensen and Mazumder, 1994). Among them, SA and GA are the latest and most promising techniques. SA is widely used in the domain of both digital and analog (Cohn et al., 1994; Plas et al., 2001) circuits. Although it yields good placement solutions, it is a very time-consuming process. In contrast, GA has been mainly applied for digital circuits. This paper describes a GA application in analog circuit placement. It is organized as follows. Section 2 introduces the design flow with the use of DesignAssistant in which this placement approach is included. Section 3 describes the implementation of this adaptive strategy. Section 4 gives the parameter optimization with fractional factorial experiment and meta- GA. Section 5 shows experimental results and the conclusion is drawn finally.

2 DesignAssistant

A design environment called DesignAssistant (Wolf et al., 1998) is provided in ALADIN which eases analog designers for their silicon compilation. The DesignAssistant is integrated in a commercial design framework. Its Graphical User Interface (GUI) executes external programs, such as the module generator, the placer and the router, to create layouts automatically. In Fig. 1 the design flow in the DesignAssis-

Correspondence to: L. Zhang
(lihong@ipe.et.uni-magdeburg.de)

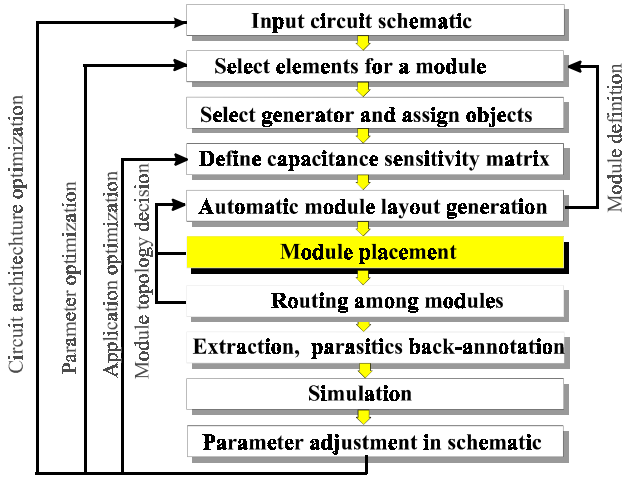

Fig. 1. Design Flow with DesignAssistant.

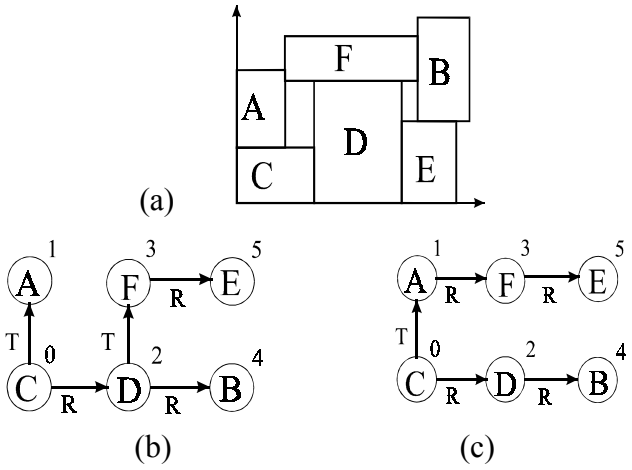

Fig. 2. Representation of BLGA with one phenotype **(a)**, the corresponding two genotypes **(b) (c)**.

tant is illustrated, where the shadowed block is the subject of this paper.

3 Algorithm implementation

The GA is a search strategy based on the mechanics of natural selection and natural reproduction in a biological system. It differs from the other stochastic search techniques by being able to encode and exploit past information efficiently during a search.

3.1 Genetic encoding

The conventional chromosomal representation of the GA is based on bit-string (Shahookar and Mazumder, 1990). A GA for the two-dimensional bin packing problem has been developed by Kroeger et al. (1991). Esbensen and Mazumder (1994) used this representation for the digital circuit placement. In this paper a bottom-left GA (BLGA) for the analog module placement is developed based on comprehensive extensions and modifications of the genetic encoding and operators found in the above work (Esbensen and Mazumder, 1994; Kroeger et al., 1991). In GA a distinction is made between genotype and phenotype of an individual. Here the genetic encoding is inspired by the two-dimensional bin packing problem, which is the problem of compactly packing a number of rectangular blocks into a bin with a fixed width and infinite height in such a way that the distance from the top edge of the highest placed block to the bottom edge of the bin is minimized. The standard algorithm for this problem places each block at a time at the downmost and then at the leftmost position. The placement algorithm is based on a generalization of this scheme. The solution space considered by the algorithm is restricted to the set of all possible BL-placements.

Assume that the given problem has n cells $c_1, ..., c_n$. An example genotype with $n = 6$ cells is shown in Fig. 2 together with the corresponding phenotypes. A binary tree (V, E), $V = \{c_1, ..., c_n\}$, in which the i'th node corresponds to the cell i, representing the absolute positions of all cells. Two kinds of edges exist: top-edges E_t and right-edges E_r, so that

$$E = E_t \cup E_r, \quad E_t \cap E_r = \emptyset \tag{1}$$

Each node has at most one outgoing top-edge and at most one outgoing right-edge. All edges are oriented away from the root of the tree. Let $e_{ij} \in E$ denote an edge from c_i to c_j and let (c_i^{xl}, c_i^{yl}) and (c_i^{xr}, c_i^{yr}) denote the coordinates of the lower left and upper right corners of c_i, respectively. Then $e_{ij} \in E_t(E_r)$ means that cell c_j is placed above (or to the right of) c_i in the phenotype. That is,

$$\forall e_{ij} \in E : c_j^{yl} \geq c_i^{yr} \text{ if } e_{ij} \in E_t, \; c_j^{xl} \geq c_i^{xr} \text{ if } e_{ij} \in E_r \tag{2}$$

The tree is decoded as follows. The cells are placed one by one in a rectangular area with horizontal length W and infinite vertical length. Each cell is moved as far down and then as far left as possible. The cells are placed in ascending order according to their priorities defined by one-to-one function $P : V \rightarrow \{0...n-1\}$. Any node has higher priority than its predecessor in the tree. In Fig. 2 the priorities are indicated at the top right hand of each node. The orientation (i.e., transformation and reflection) of each cell is defined by the function $O : V \rightarrow \{0, 1, 2...7\}$, which is also part of the genotype.

3.2 Genetic operators

Given two individuals α and β, the crossover operator generates a new feasible descendant individual γ. Two proposals are given. The operations are illustrated in Fig. 3. Throughout this section, a superscript specifies which individual the marked property belongs to. The experimental results are given in Sect. 5. In the first proposal, E^r (i.e., edge set of the

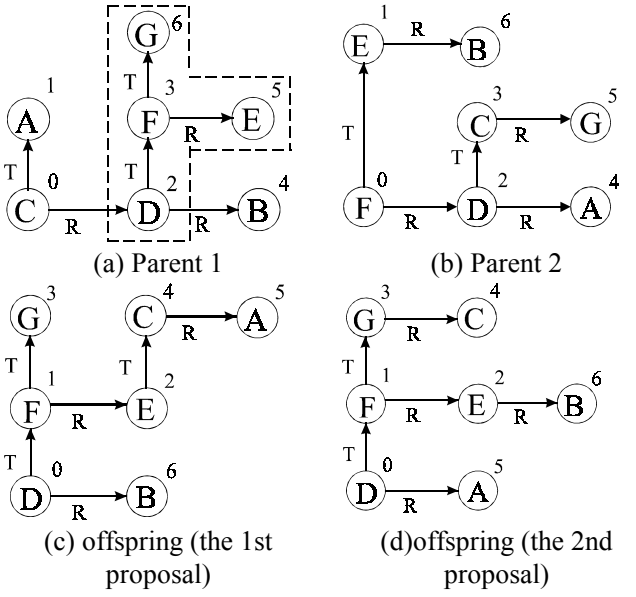

Fig. 3. Crossover operators.

descendant γ) is constructed as follows. From the cell tree of α, a connected subset

$$T_s = (V_s, E_s), V_s \subset V, E_s \subset E^{\alpha} \tag{3}$$

is chosen. T_s is chosen at random but subject to the constraint that decoding T_s in the order defined by P^{α} (i.e., priority of individual α), using

$$c \in V_s | \forall c' \in V_s / \{c\} : P^{\alpha}(c) < P^{\alpha}(c') \tag{4}$$

as root, causes no constraint violations. In Fig. 3, the chosen T_s is indicated by the dashed line. Initially E^{γ} is defined to be E_s. Hence, γ has inherited all cells in V_s from α. The remaining cells $V - V_s$ are then inherited from β by extension of E^{γ}. The cell tree of β is traversed in ascending order according to ${}^{\beta}$. At any node it is checked if the corresponding cell c belongs to V_s, that is, whether it has been placed in γ already. If so, the cell is skipped. Otherwise, c is added to the cell tree of γ by extending E^{γ} randomly. The orientation of any cell is inherited unaltered together with the cell itself. That is,

$$t^{\gamma}(c) = \begin{cases} t^{\alpha}(c) \; if \;\; c \in V_s \\ t^{\beta}(c) \; if \;\; c \in V - V_s \end{cases} \tag{5}$$

P^{γ} should correspond to the order in which the cells were placed when creating E^{γ}. Since P is a bijection, the following constraints in Eq. (6) uniquely determines P^{γ}:

$$\begin{aligned} &\forall c_i \in V_s, \;\; \forall c_j \in V - V_s : P^{\gamma}(c_i) < P^{\gamma}(c_j) \\ &\forall c_i, \;\; c_j \in V_s : P^{\alpha}(c_i) < P^{\alpha}(c_j) \Rightarrow P^{\gamma}(c_i) < P^{\gamma}(c_j) \\ &\forall c_i, \;\; c_j \in V - V_s : P^{\beta}(c_i) < P^{\beta}(c_j) \Rightarrow \\ &P^{\gamma}(c_i) < P^{\gamma}(c_j) \end{aligned} \tag{6}$$

The second proposal differs from the first one by the construction of the remaining cells $V - V_s$, which inherits from β by the ordered extension. In detail, the concatenation tries to follow the structure of β first of all. If impossible, randomly add to any free position of γ. In Fig. 3d the node B is added to the left of the node E, instead of the left of the node D in Fig. 3c. As well the node A is added to the left of the node D, instead of the left of the node C in Fig. 3c.

Five different mutation operators are developed. Each performs some random change in the given genotype. When performing each of these mutations, a part of the genotype has to be decoded to check if the mutated individual satisfies all constraints. A mutation is only performed if it does not cause any constraint violations. The purpose of the inversion operator is to weaken the linkage among genes. Given a genotype, the inversion operator computes a new genotype by rearranging the components in such a way that their mutual distances changes, while at the same time assuring that the corresponding phenotype is still the same. The inversion operator selects a subtree at random and moves it to another free position in such a way that no constraints are violated and so that the corresponding phenotype is still the same. An example of this is shown is Fig. 2b and c. This genotype tree is generated by moving the subtree rooted at the node F.

3.3 Cost function

The cost function is the goodness criterion of searched configurations. It consists of four parts, which are given in Eq. (7).

$$\begin{aligned} C = &(\alpha_{all\_area} C_{all\_area} + \alpha_{N\_area} C_{N\_area} + \alpha_{P\_area} C_{P\_area} \\ &+ \alpha_{A\_area} C_{A\_area} + \alpha_{D\_area} C_{D\_area}) + \alpha_{nets} C_{nets} \\ &+ \alpha_{asp\_rat} C_{asp\_rat} \end{aligned} \tag{7}$$

α^* is a weight factor for the corresponding cost C^*, which balances the importance of all the possible considerations according to different design requirements. The first is the area cost which is made up of the whole area, NWELL and PWELL region areas, analog and digital region areas. They will help to decrease the whole area. Moreover they could make NWELL and PWELL regions relatively concentrated in order to ease the fabrication. And analog and digital regions are separated from each other so that the constraints for mixed signal circuits can be imposed. The second is the net-length cost C_{nets}, in which priority coefficient can be specified for each net. For analog circuits the significance of different nets is distinct. Some sensitive nets, for instance differential input signal nets, should be as short as possible in order to decrease the parasitic capacitance and crosstalk. The more sensitive one net is, the higher its priority coefficient is defined. Linear or exponential operations can be chosen on the net priority. Five approaches, including half-perimeter, center-of-mass, complete graph, minimum spanning tree and minimum steiner tree, are developed for the net-length estimation so that analog designers can choose for the trade-off between accuracy and efficiency (Sait and Youssef, 1995).

```
Algorithm BLGA()
(M: the population size.)
Begin
1    input macro-cell geometry and net-list;
2    initialize the first population randomly;
3    evaluate the fitness;
4    while not (stopCriterion())
5        foreach (M * crossoverRate)
6            choose the first parent based on rank selection;
7            choose the second parent randomly;
8            do crossover to generate one offspring;
9        endfor
10       choose M individuals with the largest fitness among the
         combined set of parents and offspring;
11       foreach (M)
12           do mutation based on mutationRate;
13           do inversion based on inversionRate;
14       endfor
15   endwhile
16   output the best member;
End
```

Fig. 4. Outline of BLGA.

```
Algorithm meta-GA()
(M: the population size.)
Begin
1    set M as 20 and the generation sum as 100;
2    initialize the first population randomly;
3    evaluate the fitness;
4    while not (stopCriterion())
5        foreach (M)
6            make two random trials and select two parents from
             the population with the probability proportional to
             fitness;
7            perform crossover by selecting each parameter
             randomly from either parent with equal probability;
8            mutate offspring with 0.8 probability by selecting a
             parameter at random and adding to it a random
             number within the range of [0, 10];
9        endfor
10       choose M individuals with the largest fitness among the
         combined set of parents and offspring;
11   endwhile
12   select the fittest set of parameters from the final population;
End
```

Fig. 5. Outline of the meta-GA.

The third is the aspect-ratio cost C_{asp_rat}, which is used to control the shape of the final layout. Designers can define the ideal width-length ratio or exact width value. The more the real layout shape differs from the ideal definition, the more penalty is brought about. The graphic input window of the cost function is provided in the DesignAssistant.

3.4 Algorithm outline

The algorithm outline is depicted in Fig. 4. $stopCriterion()$ makes the evolution process terminated if no improvement has been observed for a predefined number of consecutive generations or a fixed number of generations is over. A fitness is the reciprocal of the corresponding cost.

4 Parameter optimization

Since the operators are of paramount importance to the overall performance of the algorithm, their parameters, including the population size, crossover, mutation and inversion rates, have to be investigated with care. The study of the correlation and sensitivity helps to shrink the regarded ranges and set up the exact optimal parameters in the problem of analog module placement.

4.1 Parameter analysis

The fractional factorial experiment is an important technique in Robust Design (Park, 1996). A Taguchi orthogonal array $L_{27}(3^{13})$ is employed to construct the fractional factorial experiment. The population size, crossover, mutation and inversion rates are taken as the factors. The reason why to choose such an array with 13 columns is to give a deliberate study about inter-correlation between two factors. The experiment design is depicted in Table 1, cr means crossover rate, mr means mutation rate, ir means inversion rate, M means population size and $*$ means the combination of two factors. Three columns (4, 7 and 11) are left for error estimation.

For the crossover and inversion rates, three levels are chosen as (0.2, 0.55, 0.85). (0.05, 0.1, 0.2) are chosen for the levels of the mutation rate. Two groups of experiments were performed in order to cover a wide range for the population size, one with (10, 35, 60) and the other with (40, 70, 100). The cost and execution time are taken as the search target, where the cost is the primary consideration and execution time is secondary. As a result, the correlation between parameters is found quite weak. The population size becomes insignificant to the cost if it is more than 60. The crossover and mutation rates are sensitive for the search. Even though the inversion rate is insensitive to the cost, it is 0.25 optimally with the consideration of its effect on the execution time. So the optimal value or ranges for the four parameters are set as follows. The population size was set as 60, the mutation range was from 0 to 0.1, the crossover rate from 0.55 to 1 and the inversion rate from 0.20 to 0.55.

4.2 Parameter determination

Because the levels in the orthogonal array are limited, it is rough to depend on it determining the parameter exact values. Furthermore although the result in Sect. 4.1 gives the weak correlation between two parameters, it is preferable to take the correlation into account when determining parameter exact values. So a meta-GA (Shahookar and Mazumder, 1990) depicted in Fig. 5 was employed to determine the exact rate values. The individuals in the population of the meta-GA consist of three integers in the range of [0, 10], represent-

Table 1. Design of the fractional factorial experiment

Exp. No.	1	2	3	4	5	6	7	8	9	10	11	12	13
	cr	mr	cr*mr		ir	cr*ir		mr*ir	M	cr*M		mr*M	ir*M
1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	2	2	2	2	2	2	2	2	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...
27	3	3	2	1	3	2	1	2	1	3	1	3	2

Table 2. Comparison among distinct algorithms

		SA	BLGA1	BLGA2
Circuit 1	C_{mean}	11799	11823	11698
	C_σ	176	46	39
	$T_{mean}(s)$	846	213	222
	$T_\sigma(s)$	210	40	69
Circuit 2	C_{mean}	86653	83487	82122
	C_σ	3087	2925	2165
	$T_{mean}(s)$	5068	3897	2848
	$T_\sigma(s)$	783	163	1031
Circuit 3	C_{mean}	215720	220740	216350
	C_σ	13554	4224	3932
	$T_{mean}(s)$	1143	2586	3222
	$T_\sigma(s)$	4	280	461

ing the crossover rate, inversion rate, and mutation rate of BLGA. The fitness of an individual (a BLGA with a certain parameter combination) is taken to be the fitness of the best placement that the meta-GA can find in the entire run, using these parameters. In meta-GA, the population size was 20, and the algorithm was run for 100 generations. The crossover probability was 1.

Table 3. Comparison among distinct algorithms

	circuit 1	circuit 2	circuit 3	Average
cr	0.775	0.82	0.865	0.82
mr	0.01	0.02	0.01	0.013
ir	0.235	0.41	0.41	0.352
initCost	12453	82276	220330	
endCost	11731	77992	203704	

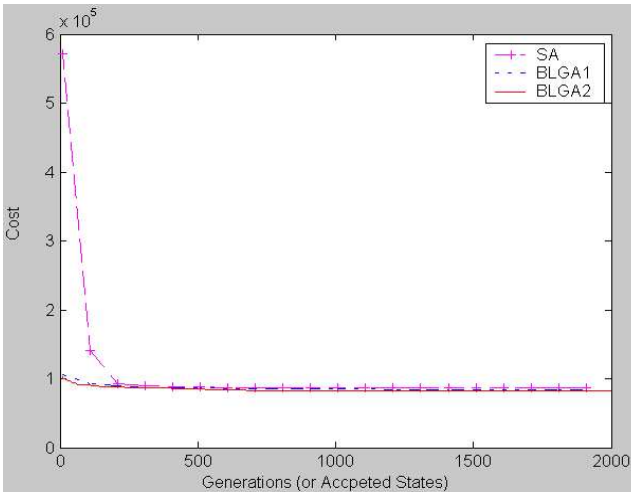

Fig. 6. Convergence schedules of different algorithms.

5 Experimental results

Because so far the analog benchmark circuits are unavailable for the synthesis purpose, three local circuits are used to evaluate the above algorithms. Each algorithm is executed for ten times so that the mean and standard deviation are used for evaluation. The cost value is superior to the execution time. As the focus here is on the comparison of algorithms, the simple half-perimeter estimation is applied for all the trials. In order to demonstrate the efficiency of GA, one optimization with SA was also performed. The results are depicted in Table 2 including SA, BLGA1 means with the first crossover proposal and BLGA2 means with the second crossover proposal.

The result shows SA is generally poorer than BLGA while need more execution time. The BLGA2 works marginally better than BLGA1. So finally the second proposal, i.e., the close inheritance crossover scheme is applied. In order to keep the diversity during the evolution, the selection based on fitness rank is applied instead of the fitness-based such as roulette wheel selection (Kroeger et al., 1991). The convergence schedule is depicted in Fig. 6. Since the costs of BLGA1 and BLGA2 are the best cost in each generation, the variance amplitude is smoother than SA in the whole view.

The representation of BLGA improves the searching efficiency so that the search need not cover a wide scope as SA but with more accuracy. The results with the Meta-GA for the three circuits are given in Table 3. Finally the best parameter set is the crossover rate of 0.82, the mutation rate of 0.023 and the inversion rate of 0.235. The program is written in $C++$ running under Solaris-UNIX in a Sun-Ultra60 workstation. The weight factors in the cost function are set as follows: $\alpha_{all\_area} = 2$, $\alpha_{N_area} = 0.2$, $\alpha_{P_area} = 0.2$, $\alpha_{A_area} = 0.2$, $\alpha_{D_area} = 0.2$, $\alpha_{nets} = 10$, and $\alpha_{asp_rat} = 5$.

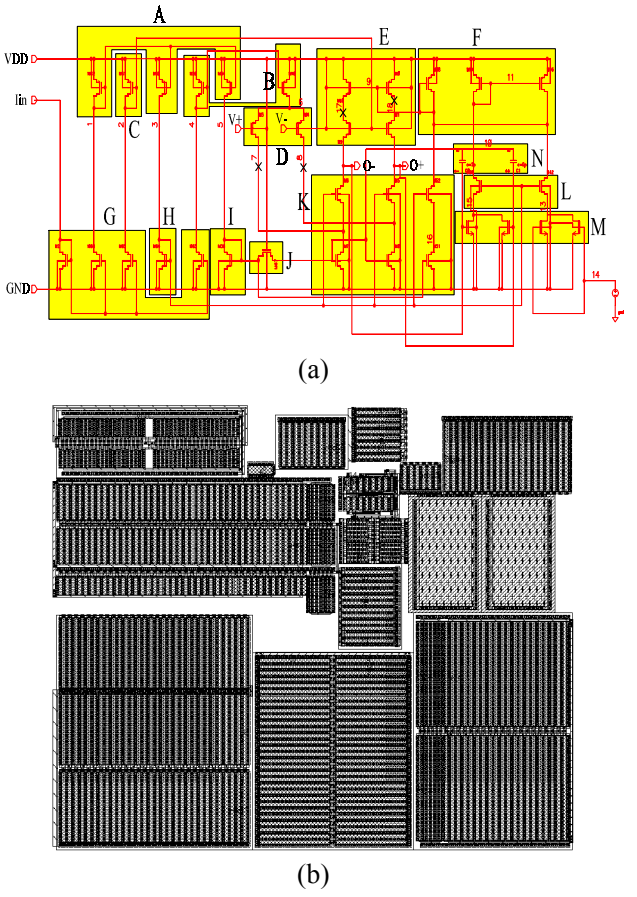

Fig. 7. Schematic **(a)** and placement layout **(b)** with BLGA2 of a common mode feedback optional amplifier.

Figure 7a gives the schematic of the third circuit, a common mode feedback optional amplifier, in which the partitioning is indicated by rectangles. The corresponding placement result with the BLGA2 is shown in Fig. 7b, which is comparable to the manual placement. In DesignAssistant the relative position of modules are recorded and passed to the router as the input. The router then compacts all the modules with the detailed routing.

6 Summary

In this paper a new technique with genetic algorithm to solve the analog module placement has been introduced. By using the notion of bin-packing, a genetic encoding has been developed in which most constraints of the problem are implicitly represented. As a consequence, each individual always satisfies constraints. The advantage of the proposed strategy is that it allows a more accurate estimation of the layout quality with the configuration space shrunk dramatically, since the use of penalty terms have been avoided. Special constraints for analog integrated circuits are included in the cost function. The fractional factorial experiment with an orthogonal array is employed in order to study the algorithm parameters. A meta-GA is applied to determine the exact parameter values. The experimental results show that this approach with the optimized parameters contributes high design efficiency with the satisfactory result, which is comparable to the manual counterpart.

References

Wang, M., Yang, X., and Sarrafzadeh, M.: Dragon2000: Fast Standard-cell Placement for Large Circuits, Proc. of the IEEE International Conference on Computer-Aided Design (ICCAD), pp 260–263, November 2000.

Zhang, L., Kleine, U., Roewer, F., Rudolph, T., and Wolf, M.: A Novel Design Tool for Analog Integrated Circuits, Proc. of the First Joint Symposium on Opto- & Microelectronic Device and Circuits, pp.146–149, April 2000.

Su, L., Buntine, W., Newton, A. R., and Peters, B. S.: Learning as Applied to Stochastic Optimization for Standard Cell Placement, IEEE Trans. on Computer-aided Design of Integrated Circuits and Systems, 20, 4, pp. 1499–1513, April 2001.

Varanelli, J. M. and Cohoon, J. P.: A two-stage simulated annealing methodology, Proc. the Fifth Great Lakes Symposium on VLSI, pp. 50–53, 1995.

Cohn, J. M., Garrod, D. J., Rutenbar, R. A., and Carley, L. R.: Analog Device-leve Layout Automation, Boston, Kluwer Academic Publishers, 1994.

van der Plas, G., Debyser, G., Leyn, F., Lampaert, K., Vandenbussche, J., Gielen, G., Sansen, W., Veselinovic, P., and Leenaerts, D.: AMGIE – A Synthesis Environement for CMOS Analog Integrated Circuits, IEEE Trans. on Computer-aided Design of Integrated Circuits and Systems, 20, 9, pp. 1037–1058, Sept. 2001.

Shahookar, K. and Mazumder, P.: A Genetic Approach to Standard Cell Placement Using Meta-Genetic Parameter Optimization", IEEE Trans. Computer-Aided Design, 9, 5, pp. 500–511, May 1990.

Esbensen, H. and Mazumder, P.: SAGA: A Unification of the Genetic Algorithm with Simulated Annealing and Its Application to Macro-Cell Placement, Proc. of The 7th International Conference on VLSI Design, pp. 211–214, 1994.

Kroeger, B., Schwenderling, P., and Vornberger, O.: Genetic Packing of Rectangles on Transputers , Transputing 91, 2, IOS Press, 1991.

Wolf, M., Kleine, U., and Schafer, F.: A Novel Design Assistant for Analog Circuits, Proc. Asia and South Pacific Design Automation Conference, pp. 495–500, Feb. 1998.

Sait, S. M. and Youssef, H.: VLSI Physical Design Automation (Theory and Practice), McGraw-Hill, 1995.

Park, S. H.: Robust Design and Analysis for Quality Engineering, Chapman & Hall, London, 1996.