A New Genetic Algorithm for the Channel Routing Problem

Jens Lienig* and K. Thulasiraman

Department of Electrical and Computer Engineering
Concordia University
1455 de Maisonneuve Blvd. West
Montreal, Quebec H3G 1M8, Canada

Abstract

A new genetic algorithm for channel routing in VLSI circuits is presented. It is based on a random path search in a lattice-like representation of the routing channel. The performance of the algorithm is tested on different benchmarks and it is shown that the results obtained using the proposed algorithm are either qualitatively similar to or better than the best published results.

1 Introduction

Channel routing of a VLSI circuit is the process of connecting pins inside a channel subject to a set of routing constraints. Its quality has a high influence on the performance and production costs of the circuit. The channel routing problem is NP-complete [1] and therefore, there is no known deterministic algorithm to solve it in a polynomial time. Hence, although different algorithms have been proposed over the years, the problem of finding the globally optimized solution for routing is still open.

New approaches are necessary to solve this problem. Such new approaches can be found in nature. Genetic algorithms guarantee, for example, the best fitness of an individual in its environment and thus, can be used as a model for mathematical optimization strategies.

To our knowledge, only three papers have been published in which strategies derived from the concept of genetic algorithms are applied to the channel routing problem [2, 3, 4]. In [2], a rip-up-and-rerouter is presented which is based on a probabilistic rerouting of nets of one routing structure. However, the routing is done by a deterministic Lee algorithm [5] and main components of genetic algorithms, such as the crossover of different individuals, are not applied. The router in [3] combines the so-called steepest descent method with features of genetic algorithms. The crossover operator, however, is restricted to the exchanging of entire nets and the mutation procedure performs only the creation of new initial individuals. The proposed algorithm in [4] is limited to the restrictive channel routing problem. Here, all vertical net segments are located on one layer and all horizontal segments are placed on the other. Furthermore, the so-called doglegs are not allowed, i.e. the horizontal segments of each net must be placed on only one horizontal row.

We present in this paper a new genetic algorithm for channel routing that is fundamentally different from the above mentioned approaches. The algorithm starts by performing a random path search to create different routing solutions of the channel. These non-optimized routing structures are seen as individuals of an initial population. Based on certain quality factors, these individuals are improved by genetic operators to eventually present a globally optimized routing result. It is shown that the resulting routing structures are either qualitatively similar to or better than the best results available in the literature.

2 Problem description

The channel routing problem is defined as follows. Consider a rectangular routing region, called *channel*, with a number of *pins* located either on the upper or the lower boundary of the channel. The pins that belong to the same net have to be connected subject to certain constraints and quality factors. The connection has to be made inside the channel on a symbolic routing area consisting of horizontal *rows* and vertical *columns*.

Three quality factors are used in this work to judge the quality of the routing result:

- minimum routing area expressed as the number of rows of the channel,
- net length and
- number of vias.

*E-mail: jensl@ece.concordia.ca

0-8186-4990-9/94 $3.00 © 1994 IEEE

3 Description of the algorithm

3.1 Survey

Genetic algorithms are optimization strategies that imitate the biological evolution process [6]. A population of individuals representing different problem solutions is subjected to genetic operators, such as, selection, crossover and mutation that are derived from the model of evolution. Using these operators the individuals are steadily improved over many generations and eventually the best individual resulting from this process is presented as the best solution to the problem.

An overview of the genetic algorithm presented in this paper is shown in Figure 1.

```
create initial population (P_c)
fitness_calculation (P_c)
p_best = best_individual (P_c)
for generation = 1 until max_generation
  P_n = ∅
  for offspring = 1 until max_descendant
    p_α = selection (P_c)
    p_β = selection (P_c)
    P_n = P_n ∪ crossover (p_α, p_β)
  endfor
  fitness_calculation (P_n)
  P_c = reduction (P_c ∪ P_n)
  p_best = best_individual (p_best ∪ P_c)
  mutation (P_c)
  fitness_calculation (P_c)
endfor
optimize (p_best)
```

Figure 1: Outline of the algorithm.

3.2 Creation of an initial population

The initial population is constructed from randomly created routing structures, i.e. individuals.

First, each of these individuals is assigned a random initial row number y_{ind}.

Let $\mathcal{S} = \{s_1, ...s_i, ...s_k\}$ be the set of all pins of the channel which are not connected yet and let $\mathcal{T} = \{t_1, ...t_j, ...t_l\}$ be the set of all pins having at least one connection to another pin. Initially $\mathcal{T} = \emptyset$. A pin $s_i \in \mathcal{S}$ is chosen randomly among all elements in $\mathcal{S}$. If $\mathcal{T}$ contains pins $\{t_u, ...t_j, ...t_v\}$ (with $1 \leq u < v \leq l$) of the same net, a pin t_j is randomly selected among them. Otherwise a second pin of the same net is randomly chosen from $\mathcal{S}$ and transferred into $\mathcal{T}$. Both pins (s_i, t_j) are connected with a so-called "random routing". Then s_i is transferred into $\mathcal{T}$. The process continues with the next random selection of $s_i \in \mathcal{S}$ until $\mathcal{S} = \emptyset$.

The creation of the initial population is finished when the number of completely routed channels is equal to the population size $|\mathcal{P}_c|$. As a consequence of our strategy, these initial individuals are quite different from each other and scattered all over the search space.

3.3 Calculation of fitness

The fitness $F(p)$ of each individual $p \in \mathcal{P}$ is calculated to assess the quality of its routing structure relative to the rest of the population $\mathcal{P}$. The selection of the mates for crossover and the selection of individuals which are transferred into the next generation are based on these fitness values.

First, two functions $F_1(p)$ and $F_2(p)$ are calculated for each individual $p \in \mathcal{P}$ according to equations (1) and (2).

$$F_1(p) = \frac{1}{n_{row}} \tag{1}$$

where n_{row} = number of rows of p.

$$F_2(p) = \frac{1}{\sum_{i=1}^{n_{ind}} (l_{acc}(i) + a * l_{opp}(i)) + b * v_{ind}} \tag{2}$$

where $l_{acc}(i)$ = net length of net i of net segments according to the preferred direction of the layer,
$l_{opp}(i)$ = net length of net i of net segments opposite to the preferred direction of the layer,
a = cost factor for the preferred direction,
n_{ind} = number of nets of individual p,
v_{ind} = number of vias of individual p and
b = cost factor for vias.

The final fitness $F(p)$ is derived from $F_1(p)$ and $F_2(p)$ in such a way that the area minimization, i.e. the number of rows, always predominates the net length and the number of vias.

After the evaluation of $F(p)$ for all individuals of the population $\mathcal{P}$ these values are scaled linearly as described in [6], in order to control the variance of the fitness in the population.

3.4 Selection strategy

The selection strategy is responsible for choosing the mates among the individuals of the population $\mathcal{P}_c$.

According to the terminology of [6], our selection strategy is actually stochastic sampling with replacement. That means any individual $p_i \in \mathcal{P}_c$ is selected with a probability

$$\frac{F(p_i)}{\sum_{p \in \mathcal{P}_c} F(p)}$$

The two mates needed for one crossover are chosen independently of each other. An individual may be selected any number of times in the same generation.

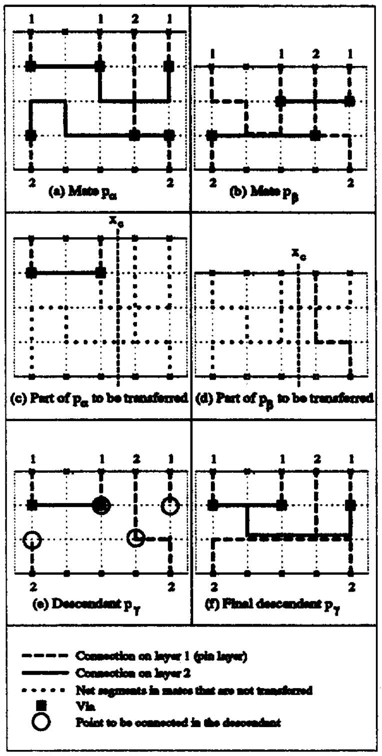

Figure 2: Crossover of (p_α, p_β) to p_γ.

3.5 Crossover operator

During the crossover, two individuals are combined to create a descendant. Let p_α and p_β be copies of the mates and p_γ be their descendant (Figure 2 (a,b)).

First, a cut column x_c is randomly selected with $1 \leq x_c < x_{ind}$, where x_{ind} represents the number of columns of the individuals.

The individual p_α (p_β) transfers the routing structure to p_γ which is located to the left (right) of the cut column x_c and not touched by x_c (see Figure 2 (c,d)).

Assume that the part of p_α (or p_β) which has to be transferred into p_γ contains rows not occupied by any horizontal segments. Then the row number of p_α (or p_β) is decremented by deleting this row until no empty row is left.

The initial row number $y_{ind\gamma}$ of p_γ is equal to the maximum of $(y_{ind\alpha}, y_{ind\beta})$. The mate which now contains less rows than p_γ is extended with additional row(s) at random position(s) before transfering its routing structure to p_γ.

The routing of the remaining open connections in p_γ is done in a random order by our random routing strategy (see Figure 2 (e,f)).

If the random routing of two points does not lead to a connection within a certain number of extension lines, the extension lines are deleted and the channel is extended at a random position y_{add} with $1 \leq y_{add} \leq y_{ind\gamma}$. If the repeated extension of the channel also does not enable a connection, p_γ is deleted entirely and the crossover process starts again with a new random cut column x_c applied to p_α and p_β.

The crossover process of creating p_γ is finished with deleting all rows in p_γ that are not used for any horizontal routing segment.

3.6 Reduction strategy

Our reduction strategy simply chooses the $|\mathcal{P}_c|$ fittest individuals of $(\mathcal{P}_c \cup \mathcal{P}_n)$ to survive as $\mathcal{P}_c$ into the next generation.

3.7 Mutation operator

Mutation operators perform random modifications on an individual. The purpose is to overcome local optima and to exploit new regions of the search space.

Our mutation operator works as follows. Define a surrrounding rectangle with random sizes (x_r, y_r) around a random centre position (x, y, z). All routing structures inside this rectangle are deleted. The remaining net points on the edges of this rectangle are now connected again in a random order with our random routing strategy.

3.8 Optimization of the best individual

The best individual, p_{best}, which has ever existed throughout the evolution process undergoes an optimization at the end of the algorithm. In this process the mutation operator is applied sequentially to p_{best}. Only improvements to p_{best} are accepted. The final p_{best} constitutes the routing solution to our specific channel routing problem.

4 Implementation and experimental results

The algorithm has been implemented in FORTRAN on a SPARC workstation. The approximate size of our source code is 8000 lines.

The performance of the algorithm has been tested on different benchmarks. The results obtained are presented in Table 1. They are compared with the best known results from popular channel routers published for these benchmarks.

In [10, Fig. 6-16], Joobbani was able to route a channel which could not be routed by the Greedy algorithm [11]. This was accomplished by using his Weaver algorithm interactively and non-interactively. As is evident from Table 1, our algorithm yields better results than the Weaver algorithm even when the latter is used interactively. Figure 3 shows our routing solution.

Benchmark	System	Col.	Rows	Netlength	Vias
Yoshimura-Kuh channel	Yosh.-Kuh [9]	12	5	75	21
	Weaver [10]	12	4	67	12
	Monreale [3]	12	4	72	11
	Our work	12	4	70	11
Joo6_12	Weaver [10]	12	4	79	14
	Packer [8]	12	4	82	18
	Monreale [3]	12	4	84	13
	Our work	12	4	79	14
Joo6_13	Greedy [11]	18	8	194	38
	Weaver [10]	18	7	169	29
	Silk [2]	18	6	171	28
	Packer [8]	18	6	167	25
	Our work	18	6	165	25
Joo6_16	Weaver [10]	11	8	131	23
	Weaver[a] [10]	11	7	121	21
	Monreale [3]	11	7	120	19
	Our work	11	6	116	15
Burstein's difficult channel	Mighty [7]	13[b]	4	83	8
	Packer [8]	12	4	82	10
	Monreale [3]	12	4	82	10
	Our work	12	4	82	8

[a] interactively

[b] empty column in the middle of the channel

Table 1: Benchmark results.

The routing results of the benchmarks are obtained by a single execution of the algorithm using an arbitrary initialization of the random number generator. No nets are prerouted and the number of layers is always defined as 2. The values of the most important parameters are as follows:

$\|\mathcal{P}_c\|$	= 50
max_descendant	= 30
max_generation	= 150
a	= 1.01 (Equation (2))
b	= 2.00 (Equation (2))
Mutation probability	= 0.001

The same parameter setting is used for all benchmarks.

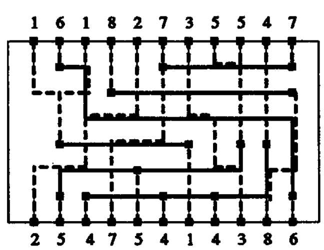

Figure 3: Our routing solution of Joo6_16.

The CPU-times of the current implementation for the benchmarks are:

Yoshimura-Kuh channel	:	5.6 min
Joo6_12	:	13.4 min
Joo6_13	:	94.2 min
Joo6_16	:	48.9 min
Burstein's difficult channel	:	9.6 min

Compared with other applications of genetic algorithms in CAD [6], our CPU-times are relatively low. Due to the inherent parallelism in genetic algorithms we are optimistic about reducing the runtime further through the implementation of a parallel version of our algorithm.

5 Conclusions

A new genetic algorithm for the channel routing problem of VLSI circuits has been presented. It has been shown that the results obtained using our algorithm are in most cases better than the best published results for channel routing benchmarks. In conclusion, genetic algorithms are very promising as approaches for solving the channel routing problem.

Acknowledgement

The authors would like to thank Mr. Henrik Esbensen, University of Michigan, Ann Arbor, for the fruitful discussions concerning this work.

References

[1] T. G. Szymanski, "Dogleg Channel Routing is NP-complete", *IEEE Transactions on Computer-Aided Design*, Vol. 4, No. 1, pp. 31-41, Jan. 1985.

[2] Y.-L. Lin et al., "SILK: A Simulated Evolution Router", *IEEE Transactions on Computer-Aided Design*, Vol. 8, No. 10, pp. 1108-1114, Oct. 1989.

[3] M. Geraci et al., "A Genetic Algorithm for the Routing of VLSI Circuits", *Euro Asic '91*, Parigi 27-31 Maggio, IEEE Computer Society Press, pp. 218-223, 1991.

[4] A. T. Rahmani and N. Ono, "A Genetic Algorithm for Channel Routing Problem", *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp. 494-498, 1993.

[5] C.Y. Lee, "An Algorithm for Path Connections and its Applications", *IRE-Transactions on Electronic Computers*, pp. 346-365, 1961.

[6] D.E. Goldberg, *"Genetic Algorithms in Search, Optimization, and Machine Learning"*, Addison-Wesley Publishing Company, New York, 1989.

[7] H. Shin and A. Sangiovanni-Vincentelli, "A Detailed Router Based on Incremental Routing Modifications: Mighty", *IEEE Transactions on Computer-Aided Design*, Vol. 6, No. 6, pp. 942-955, Nov. 1987.

[8] S. H. Gerez and O.E. Herrmann, "Switchbox Routing by Stepwise Reshaping", *IEEE Transactions on Computer-Aided Design*, Vol. 8, No. 12, pp. 1350-1361, Dec. 1989.

[9] T. Yoshimura and E.S. Kuh, "Efficient Algorithms for Channel Routing", *IEEE Transactions on Computer-Aided Design*, Vol. 1, No. 1, pp. 25-35, Jan. 1982.

[10] R. Joobbani, *"An Artificial Intelligence Approach to VLSI Routing"*, Kluwer Academic Publishers, Boston, 1986.

[11] R. L. Rivest and C. M. Fiduccia, "A Greedy Channel Router", *Proc. of the 19th IEEE Design Automation Conference*, pp. 418-424, 1982.