Available online www.jocpr.com

Journal of Chemical and Pharmaceutical Research, 2014, 6(2):1-6

Research Article

ISSN : 0975-7384
CODEN(USA) : JCPRC5

An improved FPGA routing approach based on the ant colony optimization

Fei Yang

School of Computer Science and Technology, Hubei Polytechnic University, Huangshi, P.R. China

---

ABSTRACT

The routing of integrated circuits is an important procedure of the physical design after the layout. For this reason, an improved Field Programmable Gate Array (FPGA) routing approach is proposed based on the ant colony optimization. Experimental results suggest that the proposed approach is feasible and correct.

Key words: Field Programmable Gate Array; Routing Method; Ant Colony Optimization

---

INTRODUCTION

The routing of integrated circuits is an important procedure of the physical design after the layout. The layout has decided the location of modules on chips and that of pins on modules, and provided the connection information among pins through the net-list. The routing is actually to link different modules [1]. It contains two stages, namely global routing that aims to find a routing channel for each net and detail routing that aims to distribute practical channels and holes for each net [2-3]. Global routing will reasonably distribute different parts of each net to different routing channels and specifically define routing problems in each routing channel [4]. Then, routing of channels will be completed by channel routers (a kind of algorithm) during the detail routing. The routing of FPGA is a kind of global routing and the paper only focuses on existing algorithms for global routing.

GENERAL ROUTING GRAPHS

Global routing is a classical graph theory problem. Routing areas, routing capability and relations among areas can be abstracted into a routing graph. Generally speaking, routing graphs can be divided into three kinds [5].

The first is a grid graph model. It is the simplest one, in which the chip is divided into a $m \times n$ matrix with each unit being represented with one node and different nodes being linked horizontally and vertically. The global routing here is to find a path to link all nodes in the grid graph [6]. This model performs very well as to a gate-array routing with standard units. Besides, it can be extended to solving multi-layer routing problems. However, the grid graph model is not the best choice for BBL (building block) routing.

The second is a layout graph model. It can be constructed according to the layout result in which, a module is represented with a node and the two corresponding nodes will be linked if two modules are adjacent to each other. In this way, all pins are mapped into nodes. It is usually applied to modeling the routing capacity between modules, but it cannot achieve satisfactory results in estimating edge length.

The third is a model of channel cross graph. It is the most popular model that can also express information accurately. Given a layout, a channel cross graph (also called global routing graph) can be defined as G=(V,E) in which V is the set of cross-points between one channel and another. If $\forall v_i, v_j \in V$ and one channel exists between them, then v_i is adjacent to v_j, namely that there is an edge $e_{ij} \in E$ linking v_i and v_j in the graph. Pins of modules are mapped into the corresponding edges in the global layout graph and generate new nodes, in this way a global layout graph has been extended to a global routing graph with routing information.

ROUTING OF FPGA

In this paper, the grid graph model is adopted to explore the routing of FPGA. Since the problem is only considered in plane graphs, a special grid graph model, namely the set segments, forms. The following is a repetition of the mentioned problem.

K-routing segment problem: assign segments that need routing to channels of the segmented FPGA reasonably under the constraints that, the number of occupied channel segments should not be more than K and each channel segment can only be occupied by one routing segment simultaneously.

For example, in Fig 1, there are three routing segments $n1, n2, n3$ and two kinds of segmented routing channels (a) and (b). Suppose $K=1$ here and the selected segment pattern is (a). In (a), there are two routing channels which are segmented into two channels respectively. The first one is segmented between 2 and 3, and the second one between 8 and 9. Here, the 1-routing problem is actually to assign $n1, n2, n3$ to proper channel segments and they cannot occupy one segment simultaneously. Obviously, it is infeasible to assign $n1, n2, n3$ to (a), because $n1, n2$ can only be assigned to the first segment of the second channel, so $n1$ and $n2$ cannot be routed simultaneously and there is no solution to the problem. If assign $n1, n2, n3$ to (b) or change $K=1$ with $K=2$, $n1, n2, n3$ can be routed to the channel successfully, because in (b), $n1$ and $n2$ can be assigned to either the first segment of the first channel or the first segment of the second channel, and there are also two selections for $n3$. When $K=2$, namely each routing segment can occupy two segments in the channel at most at the same time, there are two routing schemes for $n1$ and $n2$, similar to the first situation, then $n1, n2, n3$ can be successfully routed to the channel.

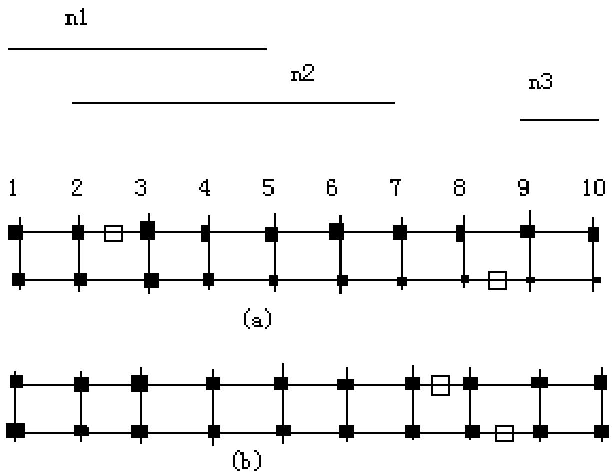

Figure 1: The sketch map of Routing problem

MAXIMUM INDEPENDEND SET PROBLEM

The FPGA routing problem can be transforms to the Maximum Independent Set (MIS) Problem.

When $K=1$, the routing of FPGA can be settled in polynomial time. When $K>1$, it is a NP-hard problem. In this paper, the problem will be first converted into a maximum independent set problem and then settled by ant colony optimization algorithms.

It has been defined that K represents the maximal number of track segments occupied at the same time by each segment that needs routing in K-routing problems.

For description, here are some symbols and their definition: the set of segments that need routing is represented with $nets$, $nets = \{net_i, i = 1, 2\cdots\}$.Routing channels are numbered from top to bottom as $1, 2\cdots$.

Feasible track: if track j is a feasible track of net_i, it means that at most K segments on j can accommodate net_j.

Duplication conflict: if net_i is in duplication conflict with $net_{i'}$, it means that crossover happens between them if they are laid on the same segment.

Segmentation conflict: if net_i and $net_{i'}$ are in segmentation conflict with each other on track j, then track j should satisfy the following three conditions:
1) j is a feasible track as to both net_i and $net_{i'}$.
2) net_i and $net_{i'}$ are not in duplication conflict.
3) if track j is assigned to net_i and $net_{i'}$, its one segment will be occupied by net_i and $net_{i'}$ in common.

According to definitions above, an undirected graph $G_k = (V, E)$ can be constructed: if track j is a feasible track of net_i, construct a vertex v_{ij} and link each pair of vertexes $(v_{ij}, v_{il})(j \neq l)$. In this way some relevant vertexes construct a group. If net_i and $net_{i'}$ are in duplication conflict with each other, introduce a symbol $J = \{ j \mid j \text{ is a feasible track as to both } net_i \text{ and } net_{i'} \}$ and link the edge $(v_{ij}, v_{i'j})(j \in J)$. If net_i and $net_{i'}$ are in segmentation conflict with each other on track j, then link the edge $(v_{ij}, v_{i'j})$.

PROPOSED APPROACH

The conversion above has transformed the MIS problem of the original graph into a minimal edge coverage problem in the corresponding line graph which is similar to TSP.

The implementation process of ant colony algorithm here is: randomly lay the k^{th} ant of the ant colony in a starting point v_i, suppose $tabu_k = \{v_i\}$, and then select edges from the edge set adjacent to v_i according to the following probability distribution:

$$P_{e_i}^k = \frac{\tau_{e_i}^{\alpha} \eta_{e_i}^{\beta}}{\sum_{e_j \in N(v_i)} \tau_{e_j}^{\alpha} \eta_{e_j}^{\beta}}$$

Here, $N(v_i)$ represents edges with non-selected nodes among those related with v_i. α and β respectively represent the importance of information and heuristic factors in the path. The heuristic factor η_{e_i} is defined as:

$$\eta_{e_i} = \frac{|ei|}{\sum_{e_i \cap e_j \neq \Phi} |ej|}, \quad e_j \in E$$

Which means that, the edge that contains more nodes and of which the neighboring edge contains fewer nodes will be selected preferentially, namely that its visibility is higher.

Put the selected edge e_i in A_k and add its nodes into $tabu_k$. Scan all edges to see whether an edge exists in which, all nodes belong to $tabu_k$, but it does not belong to A_k. Mark it and do not select it before ants start to search for the next feasible edge. Then select the next edge from the set of edges adjacent to the set $\{v \mid v \in e_i\}$ until all nodes have been covered, and finally a feasible solution will be obtained.

Pheromone τ_{e_i} is the key of the ant colony algorithm and its update is implemented as below at the end of each cycle:

$$\tau_{e_i}(t+1) = (1-\rho)\tau_{e_i}(t) + \sum_k \Delta\tau_{e_i}^k, \quad e_i \in E$$

In which:

$$\Delta\tau_{e_i}^k = \begin{cases} Q/|A_k|, & if \quad e_i \in A_k \\ 0, or & \end{cases}$$

Here, $|A_k|$ represents the number of edges contained by the minimal edge coverage found by ant k and Q is a preset positive constant.

During the ant's search process for feasible solutions, edge sets can be classified into four kinds. Ones selected as the coverage is the first kind. Both the second and third kind of edges contains overlaps with the selected edge. Get rid of the overlap part and in the rest edges, those of which the number of nodes is 1 is the second kind and those of which the number is more than 1 is the third kind. All the others belong to the fourth kind.

When the first edge is selected, the third kind of edges will be preferentially selected as its next one, and then the second one and finally the fourth kind randomly, until all nodes are have been covered.

The algorithm can be described as:
1. Initialization:
$\alpha = 2$
$\beta = 0.5$
As to each edge e_i, calculate the value of η_{e_i} according to formula (4.2), $\tau_{e_i} = 0.5$.
$t = 0.5$ (Set the temperature in Metropolis criterion)
m is equal to the number of nodes. (Suppose that the number of ants is equal to that of nodes)

The set of all nodes is point-set.

The set of all edges is edgeset.

As to each ant $\mathrm{i}, \mathrm{i} = 1,2\ldots\mathrm{m}$, $edge-set1_i = \phi$. (Initialization of the first kind of edges of each ant)
$cover-point-set_i = \varphi$ (Initialization of nodes covered by each ant)
$L_{best} = n$ (The number of edges contained by the optimal solution and its initial value is the number of nodes)
$Solution = \phi$ (Initialization of the optimal solution)
$N = m/2$ (Initialization of the maximal number of algorithm iterations)
$ci = 0$ (Initial value of the number of iterations)
$ruo = 0.9$ (Volatilization degree of pheromone)
$Q = 10$ (The control parameter for the pheromone update)
2. Search a feasible solution as to each ant.
As to each ant $\mathrm{i}, \mathrm{i} = 1,2\ldots\mathrm{m}$,
While ($cover-point-set_i != point-set$)
{
Randomly select a starting point v_j from $point-set/cover-set-point_i$ and select one edge e_k from edges related to v_j according to the probability to cover v_j.
$e_k \rightarrow edge-set1_i$
$\{v_l | v_l \in e_k\} \rightarrow \mathrm{cov}er-po\mathrm{int}-set_i$
While (the newly added edges $edge-set1_i$ contain the second and the third kind of edges)
{
Mark $edge-set1_i$ with e_{new},
Select the next edge from the third kind of edges according to the probability in formula (4.1), suppose it is e_{next},
$e_{next} \rightarrow edge-set1_i$
$\{v_l | v_l \in e_{next}\} \rightarrow \mathrm{cov}er-po\mathrm{int}-set_i$
}
}

3. Update of pheromone.

As to each ant $\mathrm{i}, \mathrm{i}=1,2\ldots\mathrm{m}$,

Calculate the number of edges contained by the solution $edge-set1_i$ obtained in the second step, suppose it is L_i .

$L_{\min}=\min\{L_i|i=1,2\ldots m\}$, suppose the ant that get $L_{\min}$ is u,

If $L_{\min}<L_{best}$

{

Then $L_{best}=L_{\min}$

$Solution=edge-set1_i$

}

Decide the ants of which the pheromone should be updated according to the Metropolis criterion and then update the pheromone according to formula (4.3) and (4.4).

$ci=ci+1$

4. Termination conditions

If ($ci>T\ \&\&L_{best}$ changes during n/10 iterations)

{

$\mathrm{cov}er-po\mathrm{int}-set_i=\phi$

$edge-set1_i=\phi$

}

Or, output the optimal solution $\mathrm{edge\ set}-\mathrm{Solution}$ and the optimal value (equal to $|\mathrm{edge}-\mathrm{set}|-L_{best}$)

The algorithm ends.

RESULTS

A group of MIS problems are introduced here to verify the performance and calculation capability of ant colony algorithms. Instance graphs adopted here are selected from Bondy, Murty and Sloane's materials. All MISs of these graphs are known and it is easy to verify the global optimization performance of the algorithm.

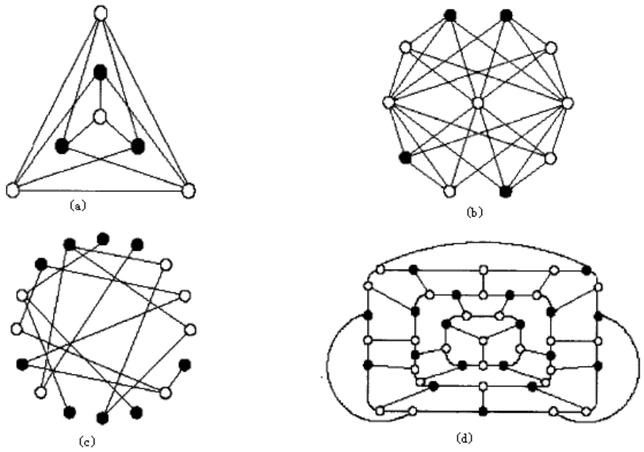

Figure 2: The examples of Maximum Independent Set

Table 1. Result of Experiments

Instance graph	Ant colony optimization algorithm				
	Number of nodes	Number of edges	MIS value	Obtained MIS	Error
Fig 2(a)	7	12	3	3	0%
Fig 2(b)	11	28	4	4	0%
Fig 2(c)	14	21	7	7	0%
Fig 2(d)	46	69	19	19	0%

Table 1 shows that the error between the result of the ant colony algorithm and the optimal solution is 0.

CONCLUSION

This paper first converted the routing problem of FPGA into a MIS problem, then further transformed it into an edge coverage problem in a super graph and finally adopted the ant colony optimization algorithm to obtain a satisfactory solution. Experimental results show that the algorithm performs well in solving MIS problems.

Acknowledgements
This work was supported by the Science and Technology Research Foundation of Education Bureau of Hubei Province (No. B2013064).

REFERENCES

[1] OY Adisa, *Construction Management and Ec.*, **2010**, 28(5), 509-526
[2] JS Shane, *j. Manage. Eng*., **2009**, 25(4), 221-229
[3] M Dorigo; A Colorni; V. Maniezzo, *Ieee T. Syst. Man. Cy.* A., **1996**, 26(1), 29-41
[4] M Dorigo; LM Gambardella, *IEEE T. Evolut. Comput*., **1997**, 1(1), 53-66
[5] LM Gambardella; ED Tailard; M Dorigo, **1999**. *J. Oper. Res. Soc*., **1999**, 50(2), 1167-1176
[6] D Costa; A Hertz, *j. Oper. Res. Soc*., **1997**, 48, 295-305