Relaxed Routing Problem with Constraint Satisfaction Problem

Saki Yamaguchi  
Department of Information and Media Engineering  
University of Kitakyushu  
Kitakyushu, Fukuoka, 808-0135  
saki.yamaugchi@is.env.kitakyu-u.ac.jp

Yasuhiro Takashima  
Department of Information and Media Engineering  
University of Kitakyushu  
Kitakyushu, Fukuoka, 808-0135  
takasima@kitakyu-u.ac.jp

Abstract— This paper proposes a relaxed routing problem formulated by constraint satisfaction problem. In the method, we utilize a one-hot representation for the routing representation and divide a three or more terminal net into several two terminal nets. We confirm the efficiency of the proposed method, empirically.

I. INTRODUCTION

Recent improvement of the device process makes the chip design more difficult. Especially, the routing must holds too many conditions. But, the existing deterministic algorithm does not have enough ability. To solve this issue, the method with constraint satisfaction problem (CSP) is promising [1]. One of the evidences is as follows: Algorithm Design Contest (ADC) in DA symposium focuses on the puzzle problem, Number-Link [2]. This problem is very similar to the routing of LSI, that is, two terminal net routing. In the contest, CSP based algorithms archive good results. Thus, CSP based algorithms have good ability to solve the difficult routing problem.

But, CSP has one large drawback. It consumes huge runtime. One of the main reasons is that CSP solver transforms CSP into another formulation, i.e., SAT problem. SAT solver becomes much faster in this decade, but it may be still slow for some case. Thus, the application of CSP to practical problem needs some acceleration method. One of the most useful solution is relaxation.

In this paper, we also focus the routing problem. As mentioned above, a subset of this problem is Number-Link. For the Number-Link, CSP can solve the problem practically. However, Number-Link is limited on only two terminal nets. In practical, netlist includes three or more terminal nets. For the practical problem, [3] reports some special problem needs huge runtime, i.e., it cannot solve the exact routing problem for a small problem within one hour. Thus, a naive application may be impractical. To solve this issue, we relax the routing problem, which permits some redundancy for the routing. We utilize CSP formulation for the relaxed routing problem. We confirm the efficiency of the proposed method, empirically. As a result, 400 times fast runtime can be obtained.

The rest of this paper is as follows: Section II introduces the proposed formulation; Section III reports the experimental results; and Section IV concludes this paper.

II. CSP BASED FORMULATION FOR RELAXED ROUTING PROBLEM

A. Relaxed Routing Problem

This paper focuses on the relaxed routing problem which formulation is as follows: The input consists of the graph and the netlist where each net is a set of vertices, called terminals; The output is the assignment of the vertices into the nets; The constraint is the assignment is exclusive and each net is completed among the assignment. The optimization objective could be sometimes set, but we consider only its feasibility. This relaxed routing problem can be considered as the routing area assignment which is widely utilized for the length matching routing in PCB routing [4], but not limited. It not only checks whether a feasible solution exists or not, but also divides the routing problem into sub-problems which searches the routing of each net. Note that the exact routing must not include redundancy, but this problem permits it. Thus, if the exact routing result is needed, we should apply an exact routing algorithm to the resultant assignment. But, we consider this realization process can be much faster than the execution without the assignment since it should solve only small and exclusive sub-problem for each net. This confirmation is in the future works.

B. Constraint Satisfaction Problem

Constraint Satisfaction Problem (CSP) consists of three elements, that is, variables, domain, and constraints. Domain corresponds to the set of values that each variable can take. Each constraint consists of a tuple of variables and a list of values that the tuple is allowed to take for the problem.

To solve CSP, several methods have been proposed. Especially, Sugar is promising [5]. Sugar translates CSP into SAT problem. Then, a SAT solver, for example, minisat [6], solves the translated problem. This framework becomes very effective due to the improvement of the performance of SAT solvers. Furthermore, the application becomes wider since its easiness to formulate the problem. For example, several puzzle problems can be solved with its formulation.

C. Application of CSP to Relaxed Routing Problem

In this paper, we propose a formulation of relaxed routing problem with CSP.
At first, we compare two following representations introduced in [7]. The first one is each vertex has one integer variable which represents the net ID. This representation ensures exclusive use of vertex implicitly since each vertex has only one variable. The other is one 0-1 variable corresponds to each vertex and each net, called one-hot representation. This representation has several variables for each vertex. Thus, we need the constraint ensuring the exclusive use. As solving it, the sum among the variables for each vertex must be at most 1. That is, $\sum_i x_{ij} \leq 1$, where $x_{ij}$ corresponds to the utilization of the vertex $i$ for the net $j$. We also consider another condition which limits the degree of each vertices is one for terminal, or zero or two for others. For the comparison between these representations, [7] reported the one-hot representation is much better than the net ID representation for Number-Link problem and [3] also reported the same result for another special routing problem. We also check their efficiencies for the relaxed routing problem.

Next, we consider three or more terminal nets. To deal with this nets, we divide them into a set of two terminal nets. Between the sub nets from the same net, the exclusive condition must be relaxed.

Fig. 1 shows an example of routing problem, where there are 1 two terminal net (net A) and 1 three terminal net (net B). For this example, the Net ID representation is illustrated in Fig. 2. In this case, each vertex has one variable whose domain is $\{A, B, \phi\}$, where $\phi$ corresponds to no assignment. On the other hand, Fig. 3 shows the case of the one-hot representation. In this case, Net B is a three terminal net. Thus, it is divided into 2 two terminal nets. Each node of each subnet has 0-1 variable. Note that the variables of net A and net B can be set to 1 exclusively, but those variables of net B can be 1 simultaneously.

To ensure the connectivity of each net, we consider the degree of each vertex. For the exact problem, the degree of terminals must be at most one, and the others must be zero or at most two. On the other hand, the relaxed problem considers only two-terminal nets. Thus, the degree of terminals must be one, and the others must be zero or two.

III. Experimental Results

To confirm the efficiency of the proposed method, we implement the above methods. The implementation environment is as follows: CSP solver is sugar [5]; SAT solver is minisat [6]; Computer consists of Intel Core i5 2.9GHz, Memory 16GB, and macOS Sierra.

At first, we compare the Net ID representation of [8] and the one-hot representation [7]. For the input data, we choose the terminals by random on grid graph. The sizes of grid graph are $5 \times 5$, $10 \times 10$, and $20 \times 20$. The number of nets of $5 \times 5$, $10 \times 10$, and $20 \times 20$ are 3, 7, and 8, respectively. All nets are two terminals only.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$5 \times 5$</td>
<td>0.752</td>
<td>1.100</td>
</tr>
<tr>
<td>$10 \times 10$</td>
<td>2.729</td>
<td>0.959</td>
</tr>
<tr>
<td>$20 \times 20$</td>
<td>--</td>
<td>1.397</td>
</tr>
</tbody>
</table>

TABLE I shows the results of runtime by Net ID and One-hot. Each element shows the runtime where ‘--’ corresponds that the runtime exceeds 10 minutes. From these results, the Net ID representation is comparable to only small data, i.e. $5 \times 5$. For the others, the One-hot representation is much faster than the Net ID. Especially, the result for $20 \times 20$ is at least 430 faster. These results follow the results in [7] and [3]. Thus, we also confirm that the one-hot representation is more efficient. Fig. 4 shows the input data of $10 \times 10$. Fig. 5 and 6 correspond to the results by Net ID representation and by One-hot representation, respectively. In the figures, square vertices correspond to the terminals. In the resultant figures, each color corresponds to each net except for gray color. While the results include the redundancy, each net can be realized with the assigned vertices.

Furthermore, we make another data of $20 \times 20$ including 7 two terminal nets and 1 three terminal net to confirm the
feasibility of the relaxation. Fig. 7 shows the input with three terminal net colored blue. Fig. 8 shows the result of Fig. 7. The runtime is 15.680 seconds.

Therefore, we confirm that the proposed method is efficient.

IV. Concluding Remarks

In this paper, we discussed the relaxed routing problem with constraint satisfaction problem. In the method, we utilized the one-hot representation for the routing presentation. We confirm that the proposed method is efficient empirically.
For the future works, we need the application to larger data and the completion of the routing from the resultant assignment.

REFERENCES


Fig. 7. Input data of $20 \times 20$

Fig. 8. Result of $20 \times 20$