Improved Method of Simulated Annealing for Unreachable Solution Space

Hiroyuki NAKANO Kunihiro FUJIYOSHI

Department of Electrical and Electronic Engineering
Tokyo University of Agriculture and Technology
2-24-16 Nakacho, Koganei, Tokyo, 184-8588 Japan
nakano@fjlab.ei.tuat.ac.jp kfujiyos@fjlab.ei.tuat.ac.jp

Abstract— Simulated Annealing is a universal probabilistic metaheuristic for optimization problems of locating a good approximation to the global minimum of given function in a large solution space. It is sometimes used for physical design problems. However, Simulated Annealing is known to be inefficient when it searches solution spaces containing infeasible solutions. In this paper, we propose two methods to make adjacent solutions for such solution spaces. Experimental comparisons indicate the effectiveness of the proposed methods.

I. Introduction

Simulated Annealing (that is called SA in the following) is a universal probabilistic metaheuristic for optimization problems of locating a good approximation to the global minimum of given function in a large solution space. If we define a method to make adjacent solutions for SA search, SA can search for an appropriate solution in the solution space[1][2].

When a solution for a problem does not satisfy a constraint of the problem, it is called an "*infeasible*" solution. If an adjacent solution obtained is infeasible in SA, it is rejected and another adjacent solution is made usually (naive method). Such SA is known to be inefficient when it searches solution spaces which include many infeasible solutions. "From any feasible solution, any other feasible solution can be obtained via a sequence of feasible solutions of the polynomial number of the size of the problem" is called "*reachability*". If a solution space includes many infeasible solutions, it does not have reachability sometimes.

When a solution space includes infeasible solutions, another technique to search such a solution space by SA is to impose very large cost to infeasible solutions (penalty function). However, when the technique is used, there is no guarantee that SA finds a feasible solution.

A special solution space of a certain problem for SA is proposed [3], where an adjacent solution can be obtained with avoiding infeasible solutions in $O(n^2)$ time and the reachability from any solution to any other is proved in this solution space. However, such method is applicable to a very limited problem, and it is not universal.

On the other hand, an efficient method to make adjacent solutions is proposed for SA by Tezuka[4]. If we use this method to search in a solution space that is not reachable, we can expect that the method makes the solution space reachable. However, the number of adjacent solutions to be made is determined only by probability, and it is independent of whether an adjacent solution is feasible or not. Therefore, we think that the efficiency of SA searching may not be good.

In this paper, we propose two methods to make adjacent solutions. One is the improved method of Tezuka's for a solution space which includes infeasible solutions, and the other is the method when a solution space includes so many infeasible solutions that Tezuka's and our first method can't search efficiently. Both proposed methods consider whether an adjacent solution is feasible. We will discuss these methods on two problems whose solution spaces include many infeasible solutions. One is the packing problem to pack rectilinear blocks, and the other is the placement problem for the fixed-die. The solution spaces of these problems include infeasible solutions, and are used to ensure effectiveness of the proposed methods. And we compare proposed methods with conventional methods by computer experiments.

II. Simulated Annealing

The idea underlying SA was proposed by Metropolis et al. in 1953[5]. The phenomenon of annealing is a process to obtain crystals with few defects from the material in a molten state at a high temperature. Applicability of the simulation to search for a good solution in the optimization problem was demonstrated by Kirkpatrick et al.[1]. Thereafter, SA was shown to be an effective method for solving combinatorial optimization problems[6].

A. SA Algorithm[7]

At each step of this simulation algorithm, a new state of the system is constructed from the current state by giving a random displacement to a randomly selected particle. If the energy associated with this new state is lower than the energy of the current state, the displacement is accepted, that is, the new state becomes the current state. If the new state has energy higher by ΔE, it becomes the current state with probability

$$\exp(-\frac{\Delta E}{k_B T}),$$

where k_B is Boltzmann's constant and T is the absolute temperature.

The only requirement for solving a combinatorial optimization problem with annealing is that there is a set of MOVEs that can transform one solution (state) into another. Preferably, the scores of such solutions are close. So, almost any combinatorial problem can be formulated as an annealing problem. However, not all problems can be solved equally efficient.

B. Simulated Annealing and solution space[8]

If adjacent solutions are not defined well, SA search becomes inefficient. Therefore, the solution space is preferable to satisfy the following property.

Reachability : From any feasible solution S_0, any other feasible solution S_ℓ can be obtained via a sequence of feasible solutions $S_1, S_2, \cdots, S_{\ell-1}$, where S_{i+1} is an adjacent solution of S_i, and ℓ is the polynomial number of the size of the problem.

If a solution space does not have reachability, SA search may not be able to reach the optimal or near-optimal solution, because it is hampered by infeasible solutions. In this case, merit of SA that can obtain a good solution according to the time is lost.

III. Method to make adjacent solutions

In this paper, we redefine "making adjacent solutions" as multiple times of perturbations in succession, in the same way as Tezuka's[4], where "perturbation" is defined as a MOVE, i.e. a single operation.

A. Geometric method

Construction of the solution space using geometric progression for determining the times of perturbations was proposed by Tezuka[4]. It is defined as follows.

Geometric method

One operation of making adjacent solutions is defined as k times of perturbations with probability P,

$$P = \begin{cases} r^{k-1}(1-r) & (1 \le k < t) \\ r^{k-1} & (k = t) \end{cases},$$

where r is a given probability, and t is the given upper limit of the times of perturbations. This method can be done by the following algorithm.

Algorithm: **Geometric method**

Input: a current solution X_0
Output: an adjacent solution X_i
for $(i = 1, 2, \cdots, t)$\{
make an adjacent solution X_i from solution X_{i-1};
if ($r >$ (random value between 0 and 1)) break;
\} (Algorithm Geometric method Ends)

B. Proposed methods

If naive method is used to make adjacent solutions, the solution space may be unreachable when it includes many infeasible solutions. One of the advantages of the geometric method is to be able to make the solution space reachable. However, performing multiple times of perturbations, the result may be more likely an infeasible solution than the conventional method. In this paper, we propose two methods to get feasible solutions in order to search the solution space including infeasible solutions efficiently. One is "Rollback method", and the other is "Feasible method". These methods are shown in the following.

Algorithm : **Rollback method**

Input: a current solution X_0
Output: an adjacent solution X_i
for $(i = 1, 2, \cdots, t)$\{
make an adjacent solution X_i from solution X_{i-1};
if ($r >$ (random value between 0 and 1)) break;
\}
while $(i > 0)$ \{
if (X_i is feasible) break;
$i--$;
\} (Algorithm Rollback method Ends)

Algorithm : **Feasible method**

Input: a current solution X_0
Output: an adjacent solution X_i
i=0;
do\{
$i++$;
make an adjacent solution X_i from solution X_{i-1};
\}while((X_i is feasible) or $(i = t)$)
(Algorithm Feasible method Ends)

An adjacent solution of Geometric method is a solution which is "the last solution obtained by a series of perturbations". But an adjacent solution of Rollback method is a solution which is "the last feasible solution obtained by a series of perturbations". On the other hand, an adjacent solution of Feasible method is a solution which is "the first feasible solution obtained by a series of perturbations".

We use two problems whose solution spaces include infeasible solutions to examine the proposed methods. One is "Rectilinear Block Packing problem" (in IV), and the other is "Placement Problem for fixed-die" (in V).

IV. Packing Problem

One of the problems for which SA is often used is the rectangle packing problem that is to pack set of rectangular blocks in the smallest rectangle area without overlapping. In this paper, we use sequence-pair as representation for packing.

A. Sequence-Pair (seq-pair) [9]

A sequence-pair (seq-pair) is an ordered pair of Γ_+ and Γ_-, where each of Γ_+ and Γ_- is a permutation of

names of given n rectangular blocks. For example, (Γ_+; Γ_-)=($abcd$; $bdac$) is a seq-pair of block set $\{a, b, c, d\}$. A seq-pair imposes a "horizontal/vertical(H/V) constraint" on every block pair as follows. For every block pair $\{a, b\}$, a is in the left of b (equivalently, b is in the right of a) if (Γ_+; Γ_-) = ($\cdots a \cdots b \cdots$; $\cdots a \cdots b \cdots$). Similarly, a is below b (equivalently, b is above a) if (Γ_+; Γ_-) = ($\cdots b \cdots a \cdots$; $\cdots a \cdots b \cdots$).

A.1. Method to make adjacent solution

In this paper, "Full-Half-Half", that is known to be a good method to make adjacent solutions for sequence-pair, is used as a perturbation.

Full-Half-Half : Select one of these exchange operations randomly and execute it: Γ_+ Half-exchange or Γ_- Half-exchange or Full-exchange.

Since seq-pair consists of a pair of permutations, its conversion operation is a permutation transformation. Each exchange operation is defined in the following.

- Γ_+ **Half-exchange :** Exchange two elements that are selected randomly in Γ_+.
- Γ_- **Half-exchange :** Exchange two elements that are selected randomly in Γ_-.
- **Full-exchange :** Exchange two elements that are selected randomly in Γ_+ and Γ_-.

In the following, "(a, b) Γ_+" denotes Γ_+ Half-exchange of elements $\{a, b\}$.

B. Rectilinear Block Packing[10]

In this paper, rectilinear block packing problem is used as a problem whose solution space includes infeasible solutions.

A polygonal block whose outside frame consists of horizontal and vertical lines only is called a "*rectilinear block*". In [10], a seq-pair based method to represent rectilinear block packing was proposed. As only rectangle blocks can be handled by seq-pair basically, each rectilinear block is partitioned into rectangles by horizontal and/or vertical lines. These rectangular blocks are called "*sub-blocks*". *Sub-blocks* are treated as rectangular blocks in seq-pair.

The decode procedure of the representation is as follows. First, H/V constraint graphs are made from seq-pair in the similar way to [9], and the weight of each vertex moves to all of its output edges. Finally, special edges called "relative position edge pair" are added to the graphs, in order to pack and align simultaneously. They make the graphs cyclic. Since the longest path algorithm for DAG can't be used, the shortest path algorithm of Ford's is applied to calculate the longest path length and obtains the coordinates of rectangles. Therefore, the time complexity of this method is $O(n^3)$, where n is the number of sub-blocks.

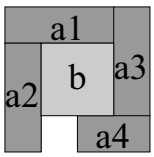


Fig. 1. the most dense packing of a rectilinear block a and a rectangular block b, corresponding seq-pair is S_{opt}=($a_1\ a_2\ b\ a_3\ a_4$; $a_2\ a_4\ b\ a_1\ a_3$)

TABLE I
THE RESTS OF CANDIDATES THAT IS OBTAINED FROM S_{opt} BY AN EXCHANGE

operation	seq-pair	feasible?
$(b, a_2)\Gamma_+$	($a_1\ b\ a_2\ a_3\ a_4$; $a_2\ a_4\ b\ a_1\ a_3$)	infeasible
$(b, a_3)\Gamma_+$	($a_1\ a_2\ a_3\ b\ a_4$; $a_2\ a_4\ b\ a_1\ a_3$)	infeasible
$(b, a_4)\Gamma_-$	($a_1\ a_2\ b\ a_3\ a_4$; $a_2\ b\ a_4\ a_1\ a_3$)	infeasible
$(b, a_1)\Gamma_-$	($a_1\ a_2\ b\ a_3\ a_4$; $a_2\ a_4\ a_1\ b\ a_3$)	infeasible

B.1. Feasible Sequence-Pair[10]

If all packing objects are rectangles, at least one packing that keeps H/V constraints imposed by any seq-pair always exists. However, if packing objects are rectilinear blocks, corresponding packing, that keeps H/V constraints imposed by a certain seq-pair, does not exist. Such seq-pair is infeasible, and called "infeasible seq-pair". In addition, seq-pair that has a corresponding packing is called "feasible seq-pair".

B.2. Unreachability of solution space

When we search solutions of rectilinear block packing problem with SA, the solution space does not have reachability often.

Focus on the most dense packing of rectilinear block a (that consists of four sub-blocks a_1, a_2, a_3, a_4) and a rectangular block b as shown in Fig.1. The placement is represented by the unique seq-pair S_{opt} = ($a_1\ a_2\ b\ a_3\ a_4$; $a_2\ a_4\ b\ a_1\ a_3$). Now, let's consider reachability to S_{opt} from a feasible seq-pair. We will check a feasible seq-pair that can be obtained from S_{opt} by an exchange because of the reversibility of Full-Half-Half. We use Full-Half-Half, so an exchange operation is Γ_+ Half-exchange or Γ_- Half-exchange or Full-exchange. However, if we use Full-exchange for any elements, the seq-pair becomes infeasible. Moreover, if we exchange two elements corresponding to sub-blocks that are made from a rectilinear block, the seq-pair becomes infeasible too. If we exchange two elements which are not adjacent in Γ_+ or Γ_-, the seq-pair becomes infeasible too, and so two elements to be exchanged have to be adjacent. Table.I shows the rests of candidates that is obtained from S_{opt} by an exchange. This table indicates that this solution space does not have reachability because any seq-pair adjacent to S_{opt} is infeasible. Hence, when the perturbation is Full-Half-Half, all the adjacent seq-pair of S_{opt} is infeasible. Therefore, the solution space does not have reachability.

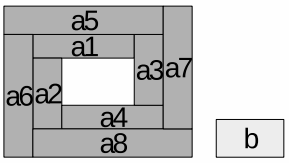


Fig. 2. input data (1): a rectilinear block a and a rectangle b

TABLE II
PROBABILITY THAT OBTAINED SOLUTIONS ARE FEASIBLE FOR THREE METHODS OF MAKING ADJACENT SOLUTIONS (INPUT DATA (1))

method	probability of feasible solutions
Full-Half-Half	1.85[%]
Geometric	0.26[%]
Rollback	2.49[%]

C. Computer Experiments

We executed two kinds of experiments which are to check for verification or comparison. We implemented SA search program based on seq-pair representation.

C.1. Verification of Rollback method

Since Rollback method is an improved method of Geometric method for the solution space including infeasible solutions, we expect that Rollback method is more effective than Geometric method in such a solution space. We executed SA search with three methods that were Full-Half-Half, Geometric method, and Rollback method.

We use the input data (1) which is made artificially as shown in Fig.2. The data consists of a rectilinear block a (that consists of eight sub-blocks $a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8$) and a rectangular block b. When b is put into a, the area of the bounding box is smallest. The common initial seq-pair was obtained by encoding the placement shown in Fig.2. The upper limit of the times of perturbations t is nine, and probability of perturbation r=0.5. Resultant probability that obtained solutions are feasible is shown in Table.II, where it is obtained by dividing the number of feasible solutions by the total times of making adjacent solutions.

From this table, when SA searches the solution space by Rollback method, feasible probability is bigger than that of Full-Half-Half and Geometric method. Therefore, we consider that Rollback method is effective to get feasible solutions in such a solution space.

C.2. Verification of Feasible method

We proposed Feasible method for a solution space which includes many infeasible solutions. Therefore, we expect that Feasible method makes such a solution space reachable. We use the input data (1) as shown in Fig.2, whose solution space has thick barrier around the densest solution and doesn't have reachability.

We executed SA search using Feasible method with several kinds of the upper limit of the times of perturbations t. Note that the method is equal to Full-Half-Half when t is one. For comparison, we executed SA search using Geometric method and Rollback method with several kinds of probability of perturbation r. The calculation time of Feasible method is greatly affected by the upper limit of times of perturbations t, and that of Geometric method and Rollback method is greatly affected by probability of perturbation r. Therefore, we adjusted the times of making adjacent solutions so that the calculation time is about 2000[sec]. To examine the probability that the final packing by SA search is the densest, SA is executed 50 times with distinct seeds of pseudorandom numbers. Table.III shows probabilities which are obtained by dividing the number of getting the densest packing by the total times of SA search.

TABLE III
PROBABILITY THAT FINAL PACKING IS THE DENSEST (INPUT DATA (1))

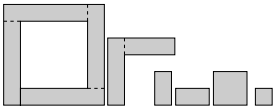
method	parameters	times of making adjacent solutions	probability
Feasible	t=1(FHH)	$6.0*10^8$	0[%]
	t=2	$2.0*10^8$	0[%]
	t=3	$1.6*10^8$	0[%]
	t=4	$1.2*10^8$	28[%]
	t=5	$1.0*10^8$	48[%]
	t=6	$8.0*10^7$	56[%]
	t=7	$7.2*10^7$	50[%]
	t=8	$6.4*10^7$	50[%]
	t=9	$6.0*10^7$	56[%]
Geometric	r=0.3	$5.7*10^8$	10[%]
	r=0.6	$5.4*10^8$	40[%]
	r=0.9	$4.9*10^8$	42[%]
Rollback	r=0.3	$2.3*10^8$	6[%]
	r=0.6	$1.5*10^8$	28[%]
	r=0.9	$7.3*10^7$	48[%]

From this table, when t is more than three, the densest packing is often obtained by SA search. Hence, we confirm that Feasible method makes the solution space reachable. When t is more than four, the probability are not very different. When Geometric and Rollback methods are used, the densest packing is sometimes obtained by SA search. However, the probabilities of the two methods are not as high as that of Feasible method. We think that Feasible method makes SA search more effective than the others when a solution space includes many infeasible solutions.

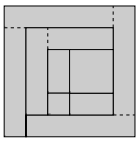
C.3. Experimental comparison for efficiency

We used four methods to make adjacent solutions in the following for comparison.

- **Full-Half-Half (FHH)**
- **Geometric method (Tezuka's method):** The upper limit of the times of perturbations t is set to the number of blocks, and probability of perturbation r=0.5.
- **Rollback method:** The parameters of this method are the same as Geometric method.
- **Feasible method:** The upper limit of the times of perturbations t is set to three.



(a)input data(2): two rectilinear blocks and four rectangles



(b)result obtained by Feasible method, packing ratio=1.0 (the densest packing of data (2))

Fig. 3. input data (2), whose solution space doesn't have reachability

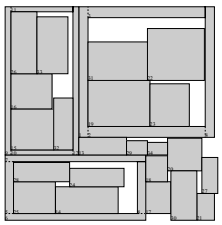


Fig. 4. input data (3), whose solution space has reachability: three rectilinear blocks and 22 rectangles. result obtained by Feasible method, calculation time=1572.1[sec], packing ratio=1.173

We used two input data which are made artificially. One consists of two rectilinear blocks and four rectangles as shown in Fig.3. The solution space does not have reachability with naive method. The other consists of three rectilinear blocks and 22 rectangles, where the result of the densest packing is shown in Fig.4. The solution space has reachability with naive method. The initial seq-pair of each problem was obtained by encoding a packing where each rectilinear block forms a row.

For each input data, we fixed the initial and final temperatures appropriately by pre-runs and carried out experiments for several cooling ratios. For each cooling ratio, SA searches are carried out five times with distinct seeds of pseudorandom numbers. Results are shown in Fig.5 and Fig.6, where the x-axis is the CPU time, and the y-axis is the packing ratio. The packing ratio is obtained by dividing the area of the bounding box by the total area of all blocks. Note that these figures do not show the process of SA. These show five final solutions obtained and the averages of them.

From Fig.5, SA search using Full-Half-Half (naive method) does not find the densest packing since the solution space does not have reachability, but SA search using the other methods can find one of the densest packing. From Fig.6, packing ratio of Full-Half-Half is better than that of the others in less than 100[sec]. But, since packing ratio of all methods are not very different in more than 100[sec], they can search the solution space which has reachability. Therefore, we consider that proposed methods are efficient and independent of whether or not the solution space has reachability.

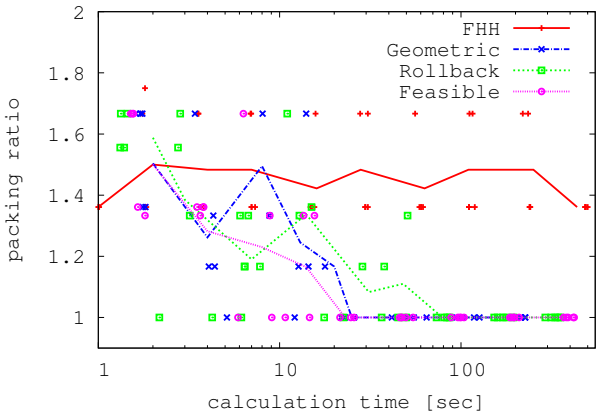


Fig. 5. comparison of search efficiency on input data (2), whose solution space doesn't have reachability

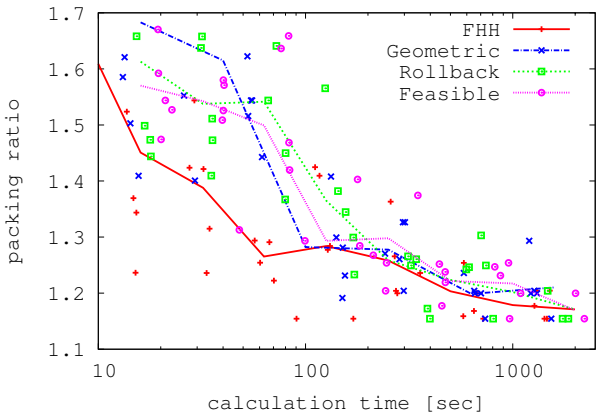


Fig. 6. comparison of search efficiency on input data (3), whose solution space has reachability

V. Placement Problem for fixed-die

One of the problems for which SA is often used is the placement problem in the LSI design, which is to place a given set of rectangular blocks into the area that is predefined size (fixed-die) so that wire length is as short as possible[11].

In other words, L_x and L_y denote the width and the height of the fixed-die respectively. X_{max} and Y_{max} denote the width and the height of a result packing respectively. Each value must meet the following requirement.

$$L_x \geq X_{max} \text{ and } L_y \geq Y_{max}$$

In the following, a solution that does not satisfy the requirement is treated as an infeasible solution.

A. Computer experiment

MCNC benchmark "ami49" was used for our experiments in this section. To fix the size of die, one of the packing results of the appropriate size is selected, and we fixed the die size as that of the packing. And the seq-pair of this packing was used as the initial seq-pair.

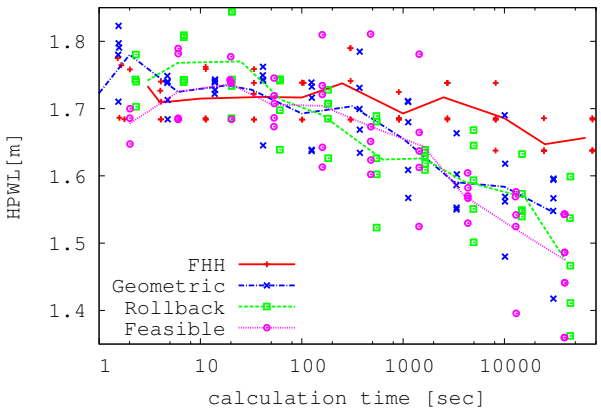


Fig. 7. comparison of search efficiency on "ami49"(49 rectangles)

The upper limit of the times of perturbations t of Feasible method is four, and the upper limit of the times of perturbations t of Geometric method and Rollback method is 49, and probability of perturbation r=0.75. Then, we fixed the initial and final temperatures appropriately by pre-runs and carried out experiments for several cooling ratios. For each cooling ratio, SA searches are carried out five times with distinct seeds of pseudorandom numbers. Result is shown in Fig.7, where the x-axis is the CPU time, and the y-axis is the HPWL, and a resultant packing is shown in Fig.8. Note that Fig.7 does not show the process of SA. It shows five final solutions obtained and the averages of them.

In this experiment, we also carry out SA search using penalty function, but the experimental results were hardly feasible. Therefore, we consider that penalty function is not suitable for such a problem whose solution space includes many infeasible solutions.

From Fig.7, we can find that proposed methods and Geometric method can make the solution space reachable for the problem with Full-Half-Half, though it is not reachable with only Full-Half-Half. In addition, proposed methods are a little more efficient than Geometric method.

From the above observations, we consider that proposed methods can make the solution space, which does not have reachability with naive method, reachable, and these are more efficient than Geometric method.

VI. Conclusion

In this paper, we proposed two methods to make adjacent solutions, which are efficient for the problem whose solution space does not have reachability. One is the improved method of Tezuka's method for a solution space which includes infeasible solutions, the other is the method when a solution space includes many infeasible solutions. And experimental comparisons indicate the effectiveness of the proposed methods.

Acknowledgements

The authors would like to thank Mr. M.Hasegawa of Tokyo University of Agriculture and Technology.

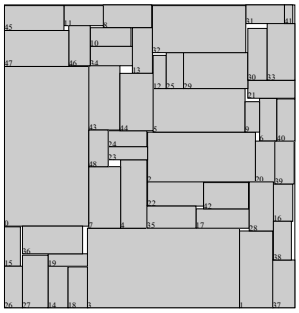


Fig. 8. placement result obtained by Feasible method for MCNC benchmark ami49, calculation time=39071[sec], HPWL=1359642[μm]

References

[1] S.Kirkpatrick, C.D.Gelatt Jr., and M.P.Vecchi, "Optimization by Simulated Annealing," Science, Vol.220, No.4598, pp.671–680, 1983.

[2] B.Hajek, "Cooling Schedules for optimal annealing," Mathematics of Operations Research, vol.13, no.2, pp.311–329, 1988.

[3] K.Kiyota, and K.Fujiyoshi, "Simulated annealing search through general structure floorplans using sequence-pair," ISCAS 2000, Vol. 3, pp.77–80, 2000.

[4] H.Tezuka, and K.Fujiyoshi, "An Effective Solution Space for Simulated Annealing," Technical report of IEICE, VLD2013–143, pp.55–60, 2014 (in Japanese).

[5] N. Metroplis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller and E. Teller, "Equation of State Calculations by Fast Computing Machines," The Journal of Chemical Physics, Vol.21, No.6, pp.1087–1092, 1953.

[6] H.Kita, "Simulated Annealing," Journal of Japan Society for Fuzzy Theory and Systems 9(6), pp.870–875, 1997 (in Japanese).

[7] R.H.J.M. Otten, and L.P.P.P. van Ginneken, "The Annealing Algorithm," Kluwer Academic Publishers, 1989.

[8] K.Fujiyoshi, T.Ohmura, and K.Ijiri, "Solution Space by Sequence-Pair Suitable for Simulated Annealing Search," Technical report of IEICE, VLD99–118, pp.9–16, 2000 (in Japanese).

[9] H.Murata, K.Fujiyoshi, S.Nakatake, and Y.Kajitani, "VLSI Module Placement Based on Rectangle Packing by Sequence-Pair," IEEE Trans. CAD, Vol.15, No.12, pp.1518–1524, 1996.

[10] K.Fujiyoshi, and H.Murata, "Arbitrary Convex and Concave Rectilinear Block Packing Using Sequence-Pair," IEEE Trans. CAD, Vol.19, No.2, pp.224–233, 2000.

[11] Y.Zhan, Y.Feng and S.S.Sapatnekar, "A fixed-die floorplanning algorithm using an analytical approach," Proc. of Asia and South Pacific Design Automation Conf, pp.771–776, 2006.