Near Optimal Solutions to the Grid Connection Problem

Sundararajan Vedantham
Department of Computer Science,
Sanjoy Das
Department of Electrical & Computer Engineering,
S. Sitharama Iyengar
Department of Computer Science,
Louisiana State University, Baton Rouge LA 70803.

Abstract *Stochastic optimization using simulated annealing has found wide applications in combinatorial optimization of NP-Complete problems in recent years, especially in VLSI related problems. The Grid Connection Problem is a combinatorial optimization problem that is proven to be NP-Complete. In this paper the simulated annealing algorithm has been applied to the Grid Connection Problem. We also propose an algorithm for the Grid Connection Problem based on certain heuristics. Results of our simulations show that while both algorithms do yield good solutions, those of the simulated annealing are superior to those of the heuristic algorithm, with rare exceptions.*

Keywords: *Combinatorial Optimization, NP-Complete problems, Simulated Annealing, Global Minima, Heuristics.*

1. Introduction

The problem of connecting a given set of points, called active points located on a grid, by means of straight lines to one of the edges of the grid is NP-Complete [1] for three or more dimensions [2]. It is also NP-Complete for two-dimensional problems when certain restrictions apply [2]. Consequently, seeking out a globally optimal solution is rendered computationally infeasible and one is motivated to look for a near optimal solution to the problem in a reasonable amount of computational time rather than perform a near-exhaustive search on the entire solution space. To the best of our knowledge, there is no algorithm that attempts to obtain such near-optimal connections. In this paper we propose two algorithms for the same purpose, which are discussed at length in sections 2 and 3. In the remainder of this paper, we shall, for the sake of conciseness, refer to the problem as the Grid Connection Problem (GCP). In this section, we provide a formal definition for the Grid Connection Problem.

An instance of the Grid Connection Problem is (i) an n-dimensional grid of size $l_1 \times l_2..l_n$, and (ii) a set of active points at specific locations in the grid, $\Sigma = \{P_i(x_1, x_2, .. x_n) | \, x_j$ is an integer and $1 \leq x_j \leq l_j.\}$. Each point in Σ is to be connected by means of a straight line to one of the several faces of the grid. An intersection of the straight line connections between any two points constitutes a violation. When portions of two straight line connections coincide for a certain (integral) length, the total number of violations is simply this length. Figure 1. shows a 6×6 grid with six active points and with violations at four different points as indicated by arrows. The objective of the GCP is to connect all active points in the grid to the sides with the least possible number of violations. The relevance of GCP in VLSI routing is immediately obvious.

2. A Heuristic Algorithm

In this section, we present a heuristic algorithm which is, in a restricted sense, greedy. Our algorithm can be applied to problems of any dimension. We observe that points that are closer to the center of the grid are potentially capable of causing a larger number of violations than those that are near the faces of the grid. Also, reducing the length of a line connecting an active point to a face should result in lower number of violations. Hence we propose the following two heuristics:

1. *Points that are closer to the center of the grid are lower in the priority list of the algorithm than those near the faces of the grid.*
2. *The algorithm has a preference for connecting points to the face of the grid that is closest.*

Within the framework of these two heuristics, the algorithm performs as follows. In the first pass it tries to connect as many points as possible without any violation. In subsequent passes it tries to connect the unconnected active points with increasing number of violations. An outline of the heuristic algorithm is provided below :

```
$\Sigma = \{P_j(x_{1j}, x_{2j},..., x_{nj}) / P_j$ is an active point.$\}$
$max = 0$
while $(\Sigma \neq \Phi)$ {
        $\forall P_j \in \Sigma$ let $(C_1^j, C_2^j,..., C_n^j) = (F, F,..., F)$
        while (another connection is possible) {
                pick $P_j \in \Sigma$ such that $\forall\ i\ x_k^i = min(x_1^j, x_2^j,..., x_n^j)$
                                        $\leq min(min(x_1^i, x_2^i,..., x_n^i))$ and $C_k^j = F$
                if (violations with $P_j$ connected along $k^{th}$ coordinate = max)
                        connect $P_j$ along $k^{th}$ coordinate
                        $\Sigma = \Sigma - \{P_j\}$
                else
                        $C_{kj} = T$
        }
        $max = max+1$
}
```

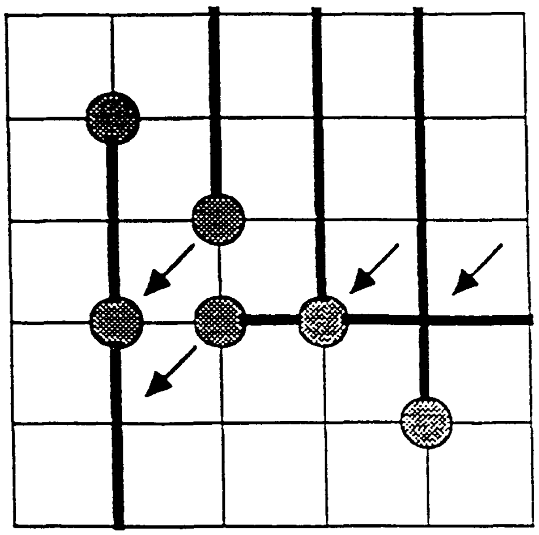

Figure 1. A 6×6 two dimensional instance of GCP with 6 active points and 4 violations.

At any stage in the algorithm, Σ is the set of unconnected active points. As points get connected, they are removed from Σ. The variable *max* contains the maximum number of violations that is permitted while connecting an active point. It is initialized to zero before the first pass of the algorithm and incremented by one at the beginning of each subsequent pass of the algorithm. This increment implies that after exhausting all the points that could be connected with a specific number of violations, the algorithm should then attempt to connect the remaining active points with one more allowable violation, in the next pass. For each unconnected active point P_j in Σ the algorithm maintains an *n*-tuple of flags $(C_1^j, C_2^j,..., C_n^j)$. At the beginning of each pass, it is updated to $(F, F,..., F)$. If the point P_j cannot be connected along a certain direction, say the *k*th direction, during a certain pass of the algorithm, the latter should then not attempt to connect it again along the same direction in the remainder of the algorithm. This is done by updating the corresponding flag C_k^j to T. When the algorithm picks a point for connection along any direction, it checks to see if the corresponding flag is F. The algorithm halts only after Σ is empty i.e. when all the active points have been connected. Having provided a formal description of the heuristic algorithm, we now provide an upper bound on its worst case time complexity.

Lemma. *The worst case time-complexity of the heuristic algorithm is $O(N^5dn)$.*

Proof: The maximum number of violations can occur in an instance of GCP only when all the active points are aligned and their straight line connections coincide for the maximum possible length. Such a case occurs only in the highly improbable situation of all the active points being in adjacent locations. If N is the number of active points and d the length of the grid along the dimension through which the connections run, then clearly the number of violations is Σ. To obtain this many number of violations, obviously the algorithm has to go through the same number of passes. In each pass of the algorithm it can perform a maximum of N iterations, each with n steps, where n is the dimensionality of the grid. Since each iteration is of $O(N^2)$, the worst case time complexity is $O(N^5dn)$. ■

Results obtained by executing the heuristic algorithm on different instances of the GCP (for various grid sizes and active points densities) are presented in Section 4.

3. Simulated Annealing for Grid Connection

The problem of connecting active points in a grid by means of straight lines is particularly suitable for simulated annealing [3] for it is trivially shown to be strongly irreducible, by the facts that it has only a finite number of possible configurations and that each configuration is attainable from any other configuration with a finite number of moves [4] [5] [6] [7] [8]. Our choice of simulated annealing as an approach is also partially motivated by the fact that it has very few requirements, and can be adapted easily to any other version of the problem merely by changing the energy function or the move selection strategy accordingly. We devote this section to a description of our approach. An outline of the well known *double loop* version of simulated annealing algorithm is presented below.

```
T = $T_{max}$
C = Initial configuration
while   (stopping condition is not satisfied) {
        for i = 1 to maximum # of iterations {
                move(C,$C_{new}$)
                $\Delta E$ = Energy($C_{new}$) - energy(C)
                if ($\Delta E < 0$) or ($e^{-\Delta E/T} >$ random(0,1))
                        C = $C_{new}$
        }
        T = $\alpha T$
}
```

3.1 Initial Configuration

The initial configuration was generated randomly. A greedy algorithm was also tried to generate the initial configuration but was later abandoned in favor of a random configuration as the former proved to be ineffective in reducing the amount of computation time.

3.2 Annealing Schedule

A good annealing schedule is of critical importance. Experimentally we found out that for an instance of GCP with n active points, a value of T_{max} numerically equal to $n/4$ itself yields the best possible result. Any higher value for T_{max} does not result in any improvement on the outcome of the algorithm, whereas any further reduction in the starting temperature results in a deterioration of the quality of the final solution.

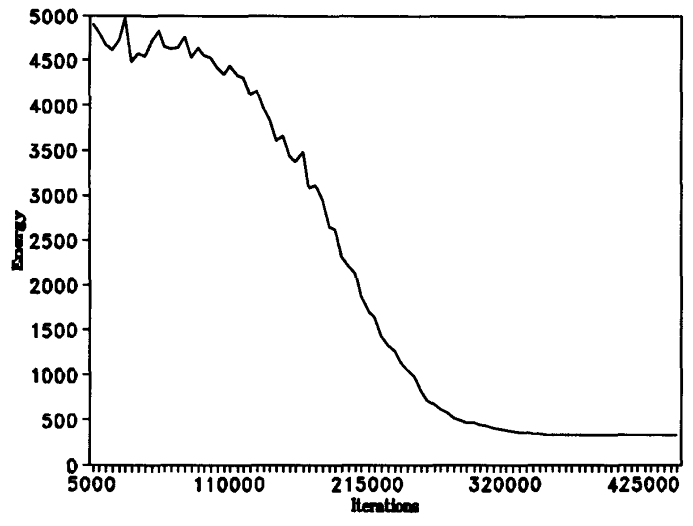

Figure 3. A sample annealing curve.

We adopted the well known *double loop* version of simulated annealing. We chose a simple geometric cooling schedule. Our results suggest that an optimal value for α (the cooling factor) is 0.9. A sample annealing curve from our simulation experiments is shown in Figure 3. We bring the algorithm to a stop when the slope of the annealing curve reaches zero, rather than basing it purely on a prefixed temperature value. Between each temperature drop we found that letting the algorithm iterate $5n$ times (where n is the number of active points on the grid) allows thermal equilibrium to be established without executing far too many iterations.

3.3 Move Generation

A move in simulated annealing results in a transition from the current configuration to a neighboring configuration within the solution space. Depending upon the control parameters, the new configuration may or may not be accepted. At each stage of the algorithm, the total number of violations is taken to be the energy of the system that needs to be minimized. Moves are generated by randomly picking an active point, and changing the face to which it is connected by rolling a die. This alteration defines the new configuration. Variation on the number of violations is calculated. A decrease in the number of violations leads to the acceptance of the move. If the move has resulted in an increase of the energy level, the move is accepted if the $e^{-\Delta E/T}$ value is greater than a random number picked between 0 and 1. It is easily seen that moves resulting in higher energy level will be accepted easily at higher temperature and the probability of such moves being accepted will continue to decrease as the temperature is decreased.

4. Simulation Results

In this section, we discuss the results of the simulations we carried out. Table 1 below, presents the results of the simulation using the heuristic algorithm.

No:	Grid Size	No. of Active points	Active Points Density %	Heuristic Algorithm No. of Violations		
				Min	Max	Avg
1	10	50	5	0	2	0.8
2		100	10	0	5	1.7
3		200	20	7	25	12.9
4		400	40	42	79	62.6
5	15	169	5	3	22	11.1
6		338	10	40	67	55.2
7		676	20	152	238	196.8
8		1352	40	553	720	631.5
9	20	400	5	43	90	68.3
10		800	10	151	259	224.7
11		1600	20	748	949	852.1
12		3200	40	3313	3819	3512.4

Table 1. Results of the simulation using the Heuristic Algorithm.

We considered grids of three different size. They were 10×10×10 (1000 points), 15×15×15 (3380 points), and 20×20×20 (8000 points). For each grid size, we ran the simulations with active points to the total grid size ratio (*active points density*) of 5, 10, 20 and 40 percent. These active points were randomly located on the grid. At each percentage level we ran the simulation ten times in order to get a clear idea of each algorithm's performance. The minimum, maximum and the average number of violations encountered for each algorithm was noted.

Table 2 below presents the results obtained for the same set of conditions by simulated annealing. The initial average number of violations is the average number of violations obtained during the initial configuration for the ten runs executed, when the active points were connected to any one of the six sides randomly. Minimum and maximum number of violations are the least and most number of violations observed at the end of the experiment among the ten trials carried out. The last column in the table presents the average number of violations observed in the final solutions of the ten trial runs. Comparing Tables 1 and 2, we observe that the heuristic algorithms performs slightly better than simulated annealing when the active points density is extremely high, whereas annealing yields much better results than the heuristic algorithm for all other cases. Table 2 also presents the details of the cooling schedule. For each configuration, we provide the average maximum and minimum temperature for each case and the average number of iterations it took to converge to the solution.

5. Conclusions

We presented two algorithms that yield a near optimal solution for the Grid Connection Problem. Experiments carried out using the algorithms show that simulated annealing performs better than the heuristic algorithm proposed in most of the cases. In cases where the active points density is very high, the heuristic algorithm performs marginally better than the simulated annealing algorithm. Since the GCP problem itself has been posed very recently, there are, to the best of our knowledge, no other algorithms available for comparison. As a direction for further research, one might consider situations where neighbouring lines connecting active points, that run parallel are also considered as violations (of a less serious nature) even though they do not intersect each other. One could also apply either algorithm to a restricted version of GCP where the active points can be connected only to a limited number of faces.

Grid Size	No. of active points	Active Points Density %	No. of Violations				Avg. Temperature		Avg. No. of Iterations
			Initial Avg	Min	Max	Avg	Initial	Final	
10	50	5	34	0	0	0	12.0	0.3694	8325
	100	10	118.2	0	0	0	25.0	0.2361	22250
	200	20	420.1	2	7	4.5	50.0	0.0758	61900
	400	40	1327.3	72	88	79	100.0	0.0640	141200
15	169	5	227.9	0	0	0	42.0	0.2051	42842
	338	10	840.5	0	6	2.8	84.0	0.0759	113399
	676	20	2716.4	63	100	82.8	169.0	0.0584	256880
	1352	40	7539.4	702	829	785.2	338.0	0.0394	587444
20	400	5	921.7	0	1	0.2	100.0	0.1172	128800
	800	10	3161.9	24	39	33.7	200.0	0.0440	323200
	1600	20	9852.9	503	539	525.3	400.0	0.0376	714400
	3200	40	25604.8	3250	3489	3357.2	800.0	0.0213	1638400

Table 2. Results of the simulation using the simulated annealing algorithm.

Bibliography

[1] M. R. Garey and D. S. Johnson, "Computers and Intractability: A Guide to the Theory of NP Completeness", *W. H. Freeman & Co., San Francisco*, 1979.

[2] Y. Birk and J. B. Lotspiech, "On Finding Non-intersecting Straight Line Connections of Grid Points to the Boundary", *IBM Technical Report*, 1989.

[3] S. Kirkpatrick, C. Gelatt and M. Vecchi, "Optimization by Simulated Annealing", *Science*, vol 14, 1983.

[4] F. Romeo and A. L. Sangiovanni-Vincentelli, "Probabilistic Hill-climbing Algorithms: Properties and Applications", *Proc. Chapel Hill Conf. on VLSI*, May 1985.

[5] E. H. L. Aarts and P. J. M. Van Laarhoven, "Statistical Cooling : A General Approach to Combinatorial Optimization Problems", *Phillips Journal of Research*, 1985.

[6] M. Metropolis, A. Rosenbluth, M. Rosenbluth, M. Teller, E. Teller, "Equation of State Calculations by Fast Computing Machines", *J. Chem Phys*, 1953.

[7] M. D. Huang, F. Romeo and A. L. Sangiovanni-Vincentelli, "An Efficient General Cooling Schedule for Simulated Annealing", *Proc. I.E.E.E. Intl. conf on Computer Aided Design*, 1986.

[8] S. R. White, "Concept of Scale in Simulated Annealing", *Proc. I.E.E.E. Intl. Conf. on Comp. Des.*, Nov. 1984.