Volume 3, No. 6, June 2012

Journal of Global Research in Computer Science

ISSN-2229-371X

REVIEW ARTICLE

Available Online at www.jgrcs.info

SIMULATED ANNEALING BASED PLACEMENT ALGORITHMS AND RESEARCH CHALLENGES: A SURVEY

Suchismita Pattanaik[1], Subhendu Prakash Bhoi[2], Rakesh Mohanty*[3]
[1]Department of Electronics and Telecomm. Engg., Veer Surendra Sai University of Technology, Burla, Odisha, India
spattanaik.vlsi@gmail.com[1]
[2]Department of Computer Science, Sambalpur University Institute of Information Technology, Burla, Odisha, India
ritzyspb.bgr@gmail.com[2]
[*3]Department of Computer Science and Engg., Veer Surendra Sai University of Technology, Burla, Odisha, India
rakesh.iitmphd@gmail.com[3]

Abstract: In this survey paper, we have done a comprehensive study of VLSI placement algorithms in the literature and classified them from different perspectives. After an extensive review, we have performed a chronological analytical study of the simulated annealing based placement algorithm. We have explored the pros and cons of simulated annealing based placement along with its applications. To the best of our knowledge, there is no state of the art survey on the simulated annealing based placement algorithms in the literature till date. This motivated us to make a survey on simulated annealing based placement algorithms. We have explored some of the research issues and challenges to open up future directions of research work in this area. We have also addressed some research issues and suggested few directions to fill up the research gap.

Key words: Electronics Design Automation, Physical Design, VLSI Design Cycle, Placement Algorithms, Simulated Annealing.

INTRODUCTION

Integrated Circuit(IC) technology is an emerging and significant development in the field of electronics. Moore's law states that, the capability of an IC has increased exponentially over the years in terms of computation power, so that the chip designers have to utilize the available area efficiently. Gradually this field of electronics is flourishing by packing more and more logic devices into the smaller and smaller areas in the circuit board. The integration of few transistors in IC is referred to as *Small Scale Integration (SSI). Very Large Scale Integration (VLSI)* is a technology that supports millions of circuit with predefined blocks and macros in a single circuit board. The VLSI concept began in 1970s, when complex semiconductor and communication technologies were being developed. The *VLSI design cycle* starts with a formal specification of VLSI chip and after undergoing through following mentioned steps, it produces a packaged chip as an output as shown in figure 1. A typical digital design follows steps such as specification of architecture, behavioral or functional design, logic design, circuit design, physical design, fabrication, packaging, testing and debugging.

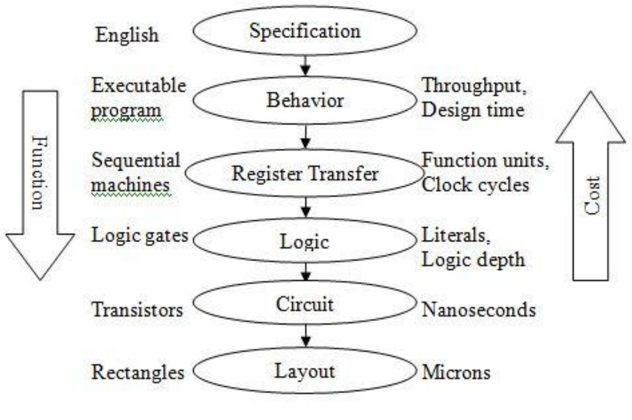

Figure 1. Design Abstraction

Placement Problem:

Placement is the part of physical design and is an essential step in electronic design automation. It assigns exact locations for various circuit components within the chip's core area. An inferior placement assignment will not only affect the chip's performance but also make it non-manufacturable by producing excessive wire length. Now we formally state the placement problem as follows [1].

Let $B_1, B_2, \ldots, B_n$ be the blocks to be placed on the chip. Each B_i , $1 \leq i \leq n$, is associated with a height h_i and a width w_i. Let $N = \{N_1, N_2, N_3, \ldots, N_m\}$ be the set of nets representing the interconnection between different blocks. Let $Q = \{Q_1, Q_2, \ldots, Q_k\}$ represent rectangular empty areas allocated for routing between blocks. Let L_i denote the estimated length of net N_i, $1 \leq i \leq m$.

The placement problem [1] is to find iso-oriented rectangles for each of these blocks on the plane denoted by $R = \{R_1, R_2, \ldots, R_n\}$ such that,

a. Each block can be placed in its corresponding rectangle, i.e. R_i has width w_i and height h_i.
b. No two rectangles overlap, i.e. $R_i \cap R_j = \phi$, $1 \leq i, j \leq n$.
c. Placement is routable, i.e. Q_j, $1 \leq j \leq k$, is sufficient to route all the nets.
d. The total area of the rectangle bounding R and Q is minimized.

Here the total wire length is minimized, i.e. is minimized. In the case of high performance circuits, the length of longest net $max \{L_i \mid i = 1, \ldots, m\}$ is minimized.

Significance of Placement and Research Motivation:

The main objective of placement is to minimize *half-perimeter wire length* (*HPWL*). A good placement can minimize the total wiring area, ensure routability, avoid signal interference, distribute heat and maximize performance. The placement that does not allow certain nets to meet their timing goals should be avoided. Due to the emerging semiconductor technologies, there is a growing trend in dealing with routing resource and constraints in the placement problem.

The placement problem is NP-complete. So finding the exact solution has been non-trivial for the researchers till date. For this reason, many heuristic approaches have been developed. Good algorithms with better data structures and sometimes the hardware implementation itself have been attempted to solve the problem. But still there is no known technique to solve the placement problem with an accurate solution.

Scope of the Survey:

One of the heuristic approaches is *simulated annealing algorithm*, which is based on the physical annealing of solid metals. This algorithm has been developed by Kirkpatrick and et al. [2]. This algorithm has gained maturity over the years and one of the most widely used algorithm applicable in many computational problems. In our work, we have performed a chronological analytical study of the simulated annealing algorithm in the context of the placement problem. To the best of our knowledge, there is no state of the art survey of the simulated annealing based placement in the literature till date. So we have made an attempt to make a state of the art literature study and explore the research issues and challenges associated with it. In this paper, we present various placement algorithms with a special focus on simulated annealing placement algorithms.

PLACEMENT ALGORITHMS

There are various algorithms that help the placers to find the near optimal solutions for the placement problem. Though the simplest greedy algorithm is fast, but it does not give satisfactory results even for simple problems. Many techniques for obtaining an exact solution require an exponential number of steps as the problem become larger. Hence the emphasis has been given towards development of heuristic techniques to solve the problem.

Slightly more complex algorithms rely on optimization to reach a local optimum by moving a single module in each iteration. Even if it improves the result, the move is preserved. At the same time, more complex variations try to move two or more labels. The algorithm ends after reaching some local optimum. Simulated annealing is a very simple algorithm that works like local optimization. However it may preserve a change of the move even if it worsens the result.

Classification of Placement Algorithms:

The placement algorithms can be classified into three broad categories as shown in figure 2. This classification is based on inputs to the algorithm, the nature of output generated by the algorithm and the techniques used by the algorithm. The algorithms which use clustering and other approaches are classified under other placement algorithms.

In another view, the modern placer can be classified into three major categories, such as Analytical approach, Min-cut partitioning based approach and hybrid approach. In hybrid approach, more than one type of algorithms is combined to develop an improved algorithm.

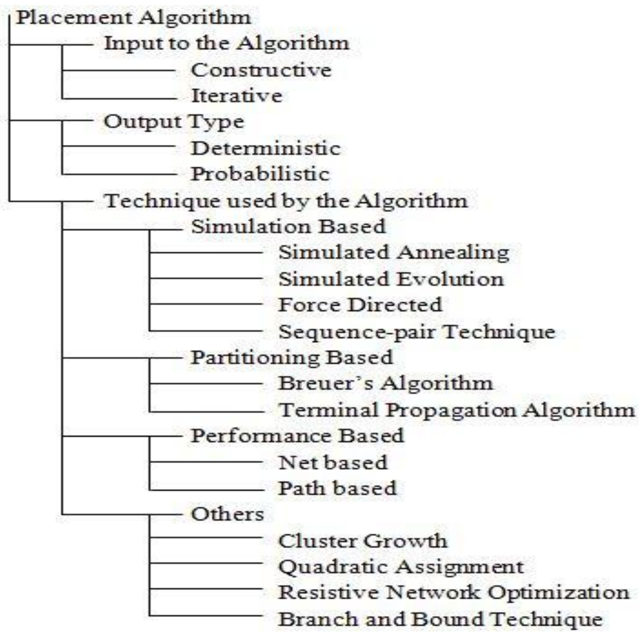

Figure 2. A Classification of Placement Algorithms

Analytical approach:

The analytical placement formulates the placement problem as mathematical program consisting of an objective function and a set of placement constraints. It optimizes the objective function through analytical approaches[3]. Generally three operations are performed in this approach as shown in figure 3.

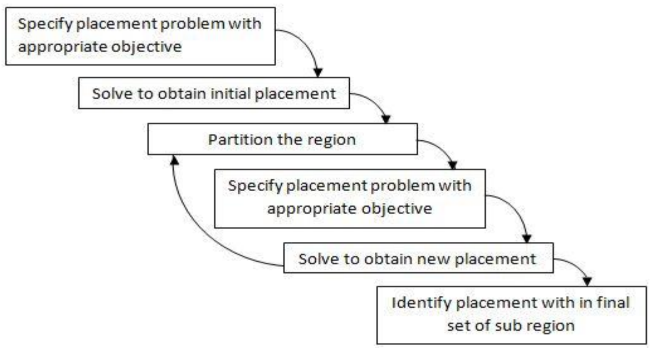
Specify placement problem with appropriate objective
Solve to obtain initial placement
Partition the region
Specify placement problem with appropriate objective
Solve to obtain new placement
Identify placement with in final set of sub region

Figure 3. Analytical Placement

First it formulates an objective function to express the placement of one or more circuit elements in a layout, then it solves the objective function to identify a placement and finally it iteratively specifies additional placement constraint and performs the first two operations until it reaches a condition for terminating the iteration.

Min Cut Based Partitioning Approach:

The min-cut placement recursively partitions the circuits and chip regions, and then assign sub-circuits into sub-regions in a top-down fashion. Since the sub-region of each module is clearly defined during placement process, the legalization of big macros can be handled pretty well. As it tries to minimize the wire length between sub-regions by minimizing the number of cuts between sub-circuits, the applicable optimization objectives are limited. It is also harder for placer of this type to handle white space in the earlier levels. Further the hierarchical approach of solving each sub problem independently might lack the global information for the interaction among different sub-regions, thus limiting the solution quality.

SIMULATED ANNEALING BASED PLACEMENT

Simulated Annealing is an optimization heuristic inspired by the cooling of solids. As a substance cools, its molecules arrange themselves into a low energy state (as measured by an energy function). The energy function could have several local minima. The cooling process is used to minimize the energy function. It is an empirical fact that the rate of cooling determines the final structure of the solid, and the solid formed by rapid cooling is usually not as stable as the solid formed by a slower cooling schedule. This is because rapid cooling forces the molecules into the first locally minimal energy state that they happen to chance upon. A slower cooling schedule increases the probability that the molecules converge on the globally minimal energy state.

Simulated Annealing (SA) Algorithm:

SA is a generic probabilistic meta-heuristic for the global optimization problem. It locates a good approximation to the global optimum in a large search space. It is often used when the search space is discrete. Its goal is merely to find an acceptably good solution in a fixed amount of time, rather than the best possible solution. The idea comes from annealing in metallurgy. This technique involves heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. Kirkpatrick applied the Metropolis algorithm to optimisation problems. We present below SA algorithm as mentioned in [3].

```
Function
        SIMULATED-ANNEALING (Problem, Schedule)
        returns a solution state.
Inputs
        Problem, : a problem.
        Schedule : a mapping from time to temperature.

Local Variables
        Current : a node,  Next :a node.
        T   : temperature controlling the probability of downward
Steps
    Current = MAKE-NODE (INITIAL-STATE [Problem])
    For  t = 1 to ∞ do
          T = Schedule[t];
       If (T = 0)
          then return Current.
         Next = a randomly selected successor of Current
         ΔE = VALUE [Next] – VALUE [Current]
        if (ΔE > 0)
          then Current = Next
          else Current = Next only with probability exp (-ΔE/T)
```

Figure 4. SA algorithm

In table 1, we have shown how physical annealing can be mapped to simulated annealing. Using these mappings any combinatorial optimization problem can be converted into an annealing algorithm.

Table 1. Physical Annealing vs. Simulated Annealing

Thermodynamic Simulation	**Combinatorial Optimisation**
System States	Feasible Solutions
Microscopic Rearrangement	Iterative Improvement
Energy	Cost
Change of State	Neighbouring Solutions
Sequence of temp. & rearrangement	Annealing Schedule
Temperature	Control Parameter
Frozen State	Heuristic Solution

Parameters of SA based Placement:

The parameters and functions used in simulated annealing algorithm determine the quality of the placement produced. An annealing algorithm requires four basic components such as *configurations*, *move set*, *cost function* and *cooling schedule* which are described as follows.

Configurations: Configurations represent the possible problem solutions for a good answer. Mathematically, it is a search a function $f(x)$ which minimizes some matrices from a set of configuration with parameters $x = (X_1, X_2, \ldots, X_N)$.

Move set: A set of legal moves may improve the current solution and easy to compute. These are also called perturbation schemes. We present three perturbation schemes as mentioned in [4].

- **a.** ***Pair wise exchange***: Let *1, 2, 3, 4, 5* is the sequence and the *integer i, j (such that* $i, j \leq n$*)* are randomly generated. Let i=1, j=4. After pair wise exchange, the new sequence is 4, 2, 3, 1, 5.
- **b.** ***Insertion scheme***: *Let i=1 and j=4*, and the sequence will be 2, 3, 4, 1, 5, by inserting the first integer in the sequence in the 4th position in the original sequence.
- **c.** ***Random insertion perturbation scheme (RIPS):*** *S={1, 2, 3, 4, 5}*. Here the digit in the 1st position can be inserted at any position to its right. Consider the 2nd position of S, i.e. 2. So 2 can be placed to its right in any position in between 3rd to nth position or and can be placed to its left in 1st position. So the sequence can be {2, 1, 3, 4, 5} or {1, 3, 4, 2, 5}.

Cost function: Cost function calculates the cost difference between the current solutions with the previous one. Then the move is accepted or rejected by Metropolis condition. If the current configuration yields better than the previous one then accept it otherwise select a random number ρ from (0,1).

$$\text{if } \rho < \exp(-(C(s') - C(s))/t), \text{ then } s = s'.$$

Cooling schedule*:* To anneal the problem from a random solution to a good solution, placement algorithm must starts initially with high temperature T. Then it accepts also the uphill moves, and it decreases at each step following some annealing schedule specified by the user.

The cooling schedule of SA consists of 4 components which are described as follows.

Starting temp: Here the initial temperature must be hot. Because at this state the uphill moves are also acceptable by the algorithm with some condition, so it maximises the search space.

Final temp: The final temperature should be close to zero, but not zero.

Temp decrement: At each temperature, the problem size is directly proportional to the number of iterations.

Iteration at each step: *Gene*rally in each temperature, a fixed number of iterations are done.

But a method, first suggested in [5] is not only to do iteration at each temperature, but to decrease the temperature *very* slowly. It is important that a large number of iterations are done at lower temperatures so that the local optimum can be fully explored.

Applications of Simulated Annealing:

Simulated Annealing have a large number of applications. It is used in VLSI design, image processing, molecular physics, job shop scheduling, event based learning situations, strategy scheduling for capital products with complex product structure, umpire scheduling in US open tennis tournament and jigsaw puzzle to name a few.

Here we discuss an application which is concerned with VLSI floor planning. According to [6], Floor planning problem can be solved both by direct solution and indirect solution with some advantages and disadvantages. In the direct solution method, the physical location of the modules on the silicon surface is rearranged and the quality of the result is measured. This approach is first implemented in [7] for gate array macro cell placement. Later it is extended for general placement problem [8]. In the indirect solution method, a graph is made by the topological representation of the modules. The graph is then annealed. Here the new arrangement gives the legal floor plan. The wire length is estimated after each move. After getting the optimal result a subsequent mapping is required. As we have to minimize the cost of the module by representing it as graph, simulated annealing algorithm is very effective in this process. An experiment was done in [9] to show how the simulated annealing algorithm is better than the KL Algorithm [10] Saab-Rao algorithm [11] by running these algorithms on same set of random graph with 50 to 500 nodes. The algorithm proposed in [9] is presented in figure 5. The algorithm uses both random and sequential neighborhood search. It uses the scheme of [5] to control the temperature because it reduces the temperature more smoothly.

Limitations of SA Algorithm and Some Future Works:

The algorithm is slow due to its iterative nature. There are some problems with simulated annealing process and they are yet to be explored. The right selection of initial position is still a big question. It may choose arbitrarily or may be the result of other algorithm. A better function to reduce the temperature very smoothly helps the algorithm to reach close to the target. For a better cooling schedule we have to select a suitable initial temperature. But till now, there is no known method for finding a suitable starting temperature for a whole range of problems. If we know the maximum distance between neighbours, then we can use this information to calculate a starting temperature. At each temperature how many iterations to be done is an issue.

Some design requires significantly less cooling time than others. For this, we have to determine a suitable termination condition which helps the algorithm to stop as soon as the placement cost is stable. The probability of accepting a worst move is based on the physical analogy and Boltzmann distribution. Again better data structure may also somehow improve the speed of the algorithm. Another challenging issue is what kind of problems can be annealed, and for those problems what is the optimal strategy for annealing them.

```
Input :
    Graph $G = (V, E)$ and the $n \times n$ cost matrix $w$.
    Cost function $f(s) = \sum_{u \in V_1} \sum_{v \in V_2} w(e_{u,v})$,
                        for any partition $s = (V_1, V_2)$.
    Starting temperature : $t_0$ ,
    Final temperature    : $t_m$
    The number of iterations : $m$.
            $\beta = (t_0 - t_m)/(m t_0 t_m)$.
Algorithm
Generate a random feasible solution $s = (V_1, V_2)$ such that $|V_1| = |V_2|$
            Let $t = t_0$, cost = $f(s)$, best_s = $s$.
            for all $v \in V_1$, enqueue ($Q_1$, $v$)
            for all $v \in V_2$, enqueue ($Q_2$, $v$)
    while $t > t_m$ do
        Let $n_1$ = dequeue ($Q_1$), $n_2$ = dequeue ($Q_2$)
        Let $s' = (V_1 - \{n_1\} \cup \{n_2\}, V_2 - \{n_2\} \cup \{n_1\})$
        Let $\Delta = f(s') - f(s)$
        Let $r$ be a random number uniformly distributed on [0, 1]
        if $\Delta \leq 0$ or $e^{-\Delta/t} > r$ then
            enqueue ($Q_1$, $n_2$), enqueue ($Q_2$, $n_1$).
            if f(best_s) > cost then best_s = $s'$.
            Let cost = $f(s')$, $s = s'$.
            Let $t = \frac{t}{1 + \beta t}$.
    end while
Output
    best_s.
```

Figure 5. SA Algorithm

A Classification of SA based Placement Algorithms:

Simulated annealing can be classified as three categories based on the techniques used as shown in figure 6. They are *Serial*, *Parallel* and *Hybrid*. Serial algorithm can be further divided into single objective and multi objective. Parallel SA placement can be classified into *error tolerance and parallelism domain* using two different criteria. Hybrid type can also be further divided into *hardware assisted* and *mimetic* respectively. The parameters are sequentially executed in serial type algorithms, whereas in parallel type algorithms, multiple parameters are tested at the same time on different processors and the best one among them is selected.

Single objective optimization have only one global optimum. In multi objective optimization, there is a set of solutions called *pareto optimal set* which are considered to be equally important and all of them constitute global optimum. *Errors* arise when parallel computations involve data dependencies. *Error tolerance* describes whether errors are prevented (e.g. by using strict synchronization schemes) or is allowed to create opportunities to improve performance. *Parallelism domain* specifies the type of parallelism exploited. The first type is *task parallel*, in which the different stages of simulated annealing are assigned to different processing units, and this is often referred to as *task decomposition* in the literature. The second type is *data parallel*. When multiple moves are made in parallel they are referred to as *parallel moves* in the literature. Since both types of parallelism are independent, so it is possible to utilize both. Hybrid algorithm is the combination of different algorithms or techniques.

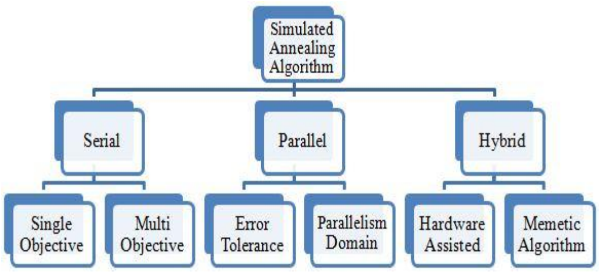

Figure 6. A Classification of SA Based Placement Algorithms

RESEARCH CHALLENGES

Simulated annealing is in NP class of problems. It gives the meta-heuristic solution. To obtain an exact solution exponential time algorithm can be used. *Parallel SA* performs many moves in parallel, thus exploring more of the configuration space in less time. But parallel moves may interact; e.g., the wire length being computed by one processor is affected by the fact that another processor is moving a module connected to this wire. The problem is that it is usually too expensive in inter-processor communication to have each move broadcasting minute details of its progress to all other active moves. Thus, we have to manage the errors that occur when parallel moves interact with each other.

By introducing the multi-objective SA algorithm, more than one objective can be considered at a single time. Since the VLSI problem usually has several objective functions, improving the SA for a multi-objective algorithm with low communication over head is a challenging issue for further investigation.

The popularity of simulated annealing has inspired several new annealing algorithms such as demon algorithm. Compressed annealing is another alternative for simulated annealing. Here the pressure and volume are also considered, in addition to temperature to address discrete optimization problems with relaxed constraints. Meta heuristic solution is achieved by simultaneously adjusting temperature and pressure in the algorithm. This is a new area of research which is emerging. Comparing the performance of simulated annealing algorithms to other local search strategies and implementing it in the field of VLSI is a major challenge for the researchers. Simulated annealing placement algorithm for standard cells has shown that the cost function does not map closely to the final area. Hence better cost functions can be devised. Hardware assisted simulated-annealing is very new area and requires substantial theoretical and experimental work before being practically used for various applications.

The most obvious direction of research in this field is implementing the hardware that helps improve the performance of the algorithm and checking the result. It is further important to combine together our fast placement solution with a hardware-assisted routing solution. Such a platform would demonstrate the feasibility of runtime placement and routing. Memetic algorithms can be developed by combining two or more search algorithms together which is also a challenging issue..

CONCLUSION

According to [6], a configuration space that is mostly smooth with gradually following hills and valleys is relatively easy to anneal. At the same time in a flat landscape where density is packed with gopher holes of different depth may stop the solution in local optima while some of the holes may contain the best solution. For this we have to rearrange the area first before applying the simulated annealing algorithm to have a better promising solution. Amongst all placers, an analytical placer NTU Place3 gives better result than others till date.

The placement problem in VLSI design is a major challenge for the placer. Many placement algorithms have been developed to find a better arrangement of circuits in a single board. In this paper, we have focused on a detailed explorative study of the simulated annealing based placement and pose some new research challenges. We have also addressed some issues to fill up some the research gaps in the literature.

REFERENCES

[1]. Naveed Sherwani, “Algorithms for VLSI physical Design Automation.” Kluwer Academic Publishers, 3rd ed. 1999.

[2]. S. Kirkpatrick, C. D. Gelatt, M.P. Vecchi, . "Optimization by Simulated Annealing". *Science* **220** (4598): 671–680., 1983.

[3]. S. Russell, P. Norvig. “Artificial Intelligence- A Modern Approach, 3rd ed. 1995.

[4]. S. Benedict, V. Vasudevan. “Scheduling of Scientific Workflows Using SA for Computational Grids”, International Journal of Soft Computing, 2(5), 606-611, 2007.

[5]. M. Lundy and A. Mess. “Convergence of annealing algorithm.” Mathematical Programming, vol. 34, pp. 111-124, 1986.

[6]. Rob A. Rutenber, “Simulated Annealing Algorithms: An Overview”. IEEE Circuits and Devices Magazine, 5(1), 19-26, 1989.

[7]. D.W. Jensen and C.D. Gelatt, Jr, “Macro Placement by Monte Carlo Annealing,” Proc. IEEE Intl Conf. on Computer Design, pp. 495-498, Nov. 1984.

[8]. C. Sechen and A.L. Sangiovanni-Vincentelli, “The TimberWolf Placement and Routing Package,” IEEE J. Solid-State Circ., vol.20, pp. 510-522, 1985.

[9]. Y.C. Zhao, L. Tao, K. Thulasiraman, M.N.S. Swamy. “An Efficient Simulated Annealing Algorithm for Graph Bi-sectioning.”Proceedings of the Symp. on Applied Computing, 65-68, 1991.

[10]. B.W. Kernighan and S. Lin. “An Efficient Heuristic Procedure for Partitioning Graphs.” The Bell Technical Journal, Vol. 49, pp. 291-307, February, 1970.

[11]. Y.G. Saab and V.B. Rao. “Fast effective heuristic for the graph bi-sectioning problem.” IEEE Trans. Computer-Aided Design, Vol. CAD-9, pp. 91-98, January, 1990.

.