# GLOBAL OPTIMIZATION ALGORITHMS FOR VLSI COMPACTION PROBLEMS

**Faiz A. Al-Khayyal***

*School of Industrial and Systems Engineering*
*Georgia Institute of Technology*
*Atlanta, Georgia 30332-0205, U.S.A.*

**Panos M. Pardalos**

*Department of Industrial and Systems Engineering*
*University of Florida*
*Gainesville, Florida 32611, U.S.A.*

## 1. INTRODUCTION

Optimization problems that appear in integrated circuit design are characterized by complex tradeoffs between multiple objectives while satisfying linear (and nonlinear) constraints. Typical of VLSI problem objectives include area, delay, and power. VLSI technology evolves very fast. Certainly there are many problems connected with VLSI technology and the main problem is in fact a multi-objective optimization problem, and therefore very difficult to solve. In this note we address only one issue in this complicated process.

Many of the optimization problems in VLSI, in their higher formulation, are nonconvex quadratic programs with either linear constraints or quadratic constraints that can be easily linearized. For a general survey of optimization techniques used in integrated circuit design see Brayton *et al.* [1]. Applications using nonconvex quadratic models can be found in Ciesieski and Kinnen [2], Kedem and Watanabe [3], Maling *et al.* [4], Soukup [5], and Watanabe [6]. A survey of algorithms for nonconvex quadratic programs can be found in a recent monograph by Pardalos and Rosen [7].

A circuit is built on the surface of a silicon substrate by interconnecting pieces of material in three primary layers:

- a conducting layer in metal which is used for electrical connections; and
- two layers of semiconductors (polysilicon and diffusion) are used for building such devices as switches, inverters, and gates as well as for electrical connections.

Layers are always insulated from each other by insulating material, and the electrical connections can be established between layers of special structures (see, *e.g.*, Mead and Conway [8] for further details).

An important optimization problem in VLSI design arises repeatedly in the process of solving the circuit compaction problem. We model this nonconvex programming problem as a jointly constrained bilinear program and present a linear programming relaxation which is equivalent to convexifying the bilinear program over a bounded region. In the overall VLSI design process, we propose using these linear programs in lieu of heuristics to determine circuit compaction. We also indicate how the bilinear programs can be solved to optimality using an available global optimization code, thereby obtaining even better VLSI designs. The purpose of this note is to indicate how some new global optimization techniques can be used in the proces of VLSI technology.

## 2. GLOBAL OPTIMIZATION FORMULATION OF THE COMPACTION PROBLEM

We consider the compaction problem in VLSI. The approach considered here is based on Lin and Allen [9].

* To whom correspondence should be addressed.

The process of solving circuit compaction problems involves three main steps (see, *e.g.*, Mead and Conway [8] or Watanabe [6]):

- translation of stick diagrams into an optimization problem;
- solution of the optimization problem; and
- modification of stick diagrams and optimization model to improve the layout.

The stick diagram is a lower level representation of an electric circuit. In a stick diagram, circuit elements such as transistors and butting contacts are represented by symbols. The interconnections among these circuit elements are represented by lines.

In the compaction problem, we minimize the area of the bounding rectangle as well as the individual rectangles in the layout. Employing this strategy iteratively, we "compress" the layout in one dimension and then the other following a user specified sequence. We can think of compaction in the horizontal direction as a process of determining the abscissas of the vertical edges of each rectangle in a given layout such that the bounding area and the layout areas are minimized.

The optimization model to be solved consists of two problems.

*Problem 1.* Minimize the area of the bounding rectangle (see Figure 1):

$$\text{Minimize } (x_R - x_L)$$
$$\text{subject to } \{\text{a set of network constraints S}\}.$$

As shown in Figure 1, in the first step we compress the layout in one dimension, which gives rise to the above linear program. The other dimension of the bounding rectangle is controlled by the designer (see [9]). The constraints in $S$ are further explained at the end of this section.

*Problem 2.* Minimize the total weighted areas of individual rectangles (see Figure 2):

$$\text{Minimize } \sum_{i=1}^{N_r} w_i(x_R^i - x_L^i)(y_U^i - y_L^i)$$

$$\text{subject to } \{\text{a set of constraints } S\}$$

$$\text{and } \quad x_R^o - x_L^o = x^o$$

where $x^o$ is the solution obtained in Problem 1, $w_i$ is the weight associated with rectangle $R_i$, $N_r$ is the number of rectangles in the layout, and $(x_L^i, y_L^i)$ and $(x_R^i, y_U^i)$ are the coordinates of the lower-left and upper-right corners, respectively, of rectangle $R_i$ for $i = 1, 2, \ldots, N_r$.

By specifying different weights to $w_i$, we can control the priority of minimization among layers. This is very useful when there is a tradeoff in area between two rectangles on different layers.

The set of constraints $S$ in this model consists of linear binary inequalities with $\pm$ 1 coefficients. These constraints include:

- minimum width and separation constraints
- connection constraints
- compaction constraints
- user specified constraints, *etc.*

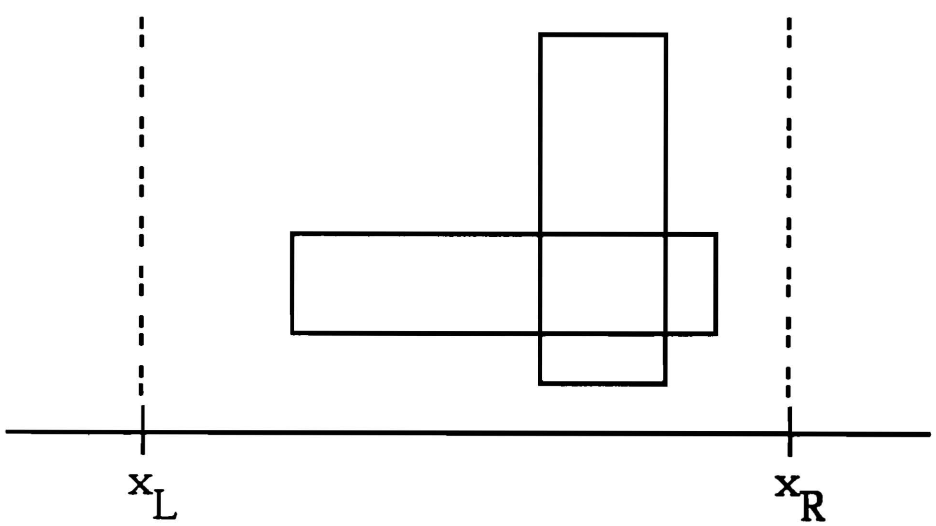


*Figure 1. Bounding Rectangle Area.*

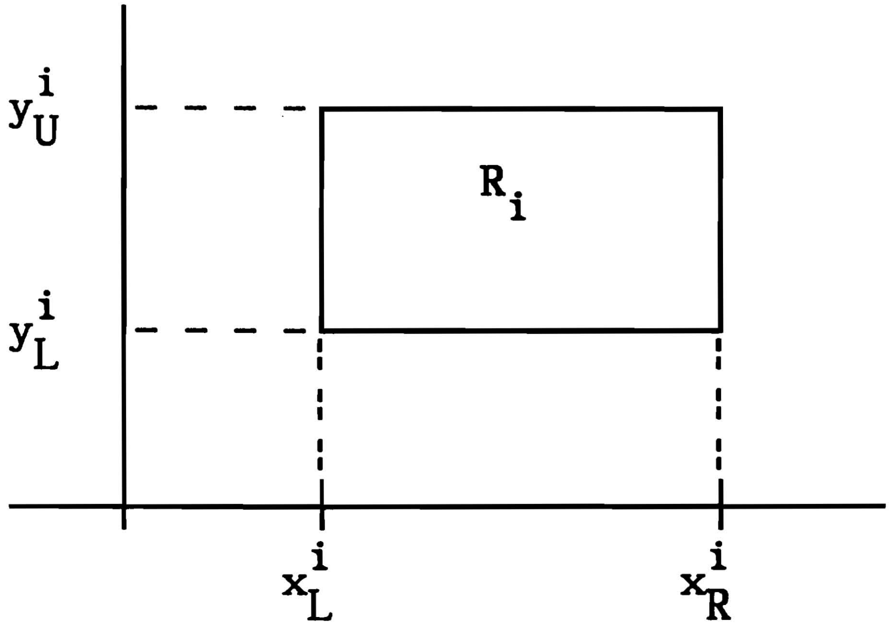


*Figure 2. Location of ith Rectangle ($R_i$).*

The minimum width constraints describe the minimum width design rules of available layers. The separation constraints describe the relationships between edges of separated rectangles, and the connection constraints describe the relationships between edges of connected rectangles. All constraints have the form $x - yRb$, where $R$ is $=$, $\leq$, or $\geq$. It is well known that the coefficients of the constraint matrix represent the adjacency matrix of a directed graph $G$, where each vertex of $G$ represents a vertical edge in the layout and each edge represents an inequality (equalities are replaced by two inequalities).

## 3. BILINEAR PROGRAMMING APPROACH

After reformulation, it is easy to see that Problem 2 can be stated as a bilinear programming problem of the form

$$\text{global min } F(x,y) = \sum_{i=1}^{n} \lambda_i x_i y_i$$

$$\text{subject to} \quad (x,y) \in P$$

where $\lambda_i$ are constants and $P$ are network constraints. This problem is a jointly constrained bilinear program that can be solved using the algorithm of Al-Khayyal and Falk [10]. The procedure in [10] assumes that all $\lambda_i > 0$, and, otherwise, $n$ additional constraints $z_i = \lambda_i y_i$ are introduced. The objective function is an instance of an indefinite quadratic form and the above problem is known to be NP-hard. The algorithm in [10] creates a sequence of successively tighter piece-wise linear functions that underestimate the objective over successively finer partitions of rectangular sets bounding all variables. The underestimating functions are *convex envelopes* (highest *convex* underestimating functions) of $x^T y$ over a specified rectangle. With minor notational changes the formulas in [10] for the convex envelope of $x^T y$ can be adjusted for $F(x,y)$ provided $\lambda_i \geq 0$ for all $i$. The case where $\lambda_i < 0$ is allowed can be similarly treated by using the *concave envelope* of $x^T y$ over a rectangle [11], Theorem 10.

We next construct the convex envelope of $F$ over a hyperrectangle. It readily follows that the convexified problem is equivalent to a linear program. Let $(l,m)$ and $(L,M)$ denote the vectors of lower bounds and upper bounds, respectively, of $(x,y)$.

These quantities can either be obtained from the solution of Problem 1, or tight bounds on the location of rectangle $R_i$ can be deduced from the constraints defining $P$. In the approach we follow, tighter bounds yield a better initial approximate solution. It is convenient to define the rectangular sets

$$\Omega = \{(x,y) \colon l \leq x \leq L,\ m \leq y \leq M\}$$

and

$$\Omega_i = \{(x_i,y_i) \colon l_i \leq x_i \leq L_i,\ m_i \leq y_i \leq M_i\}.$$

Note that $\Omega = \Omega_1 \times \Omega_2 \times \ldots \times \Omega_n$. Letting $f_A$ denote the convex envelope of a function $f$ over a set $A$, we have from Theorem 11 in [11]:

$$F_\Omega(x,y) = \sum_{i \in I^+} \max\{\lambda_i(m_i x_i + l_i y_i - l_i m_i), \lambda_i(M_i x_i + L_i y_i - L_i M_i)\} + \sum_{i \in I^-} \max\{\lambda_i(M_i x_i + l_i y_i - l_i M_i), \lambda_i(m_i x_i + L_i y_i - L_i m_i)\},$$

where $I^+ = \{i: \lambda_i > 0\}$ and $I^- = \{i: \lambda_i < 0\}$.

Since $F_\Omega(x,y) \leq (F(x,y)$ for all $(x,y)$ in $\Omega$ and the convex envelope is the highest convex function understanding $F$ over $\Omega$, then solving the following convex program will yield an approximate solution to Problem 2.

$$\text{minimize } F_\Omega(x,y)$$
$$\text{subject to } (x,y) \in P \cap \Omega.$$

But this problem is equivalent to the linear program

$$\text{minimize } \sum_{i=1}^{n} z_i$$
$$\text{subject to } (x,y) \in P \cap \Omega$$

$$\begin{array}{ll}
(1/\lambda_i)z_i \geq m_i x_i + l_i y_i - l_i m_i & i \in I^+ \\
(1/\lambda_i)z_i \geq M_i x_i + L_i y_i - L_i M_i & i \in I^+ \\
(1/\lambda_i)z_i \leq M_i x_i + l_i y_i - l_i M_i & i \in I^- \\
(1/\lambda_i)z_i \leq m_i x_i + L_i y_i - L_i m_i & i \in I^-
\end{array}$$

With the above linear program defining the subproblems, the branch-and-bound strategy presented in [10,11] can now be used to solve for the global solution of Problem 2.

The linear program above will yield an approximate solution to the problem. In the early design stages, the solution obtained from a single linear program can be used in the iterative process of compaction. Using this information only, we may reformulate the problem and see if different (user specified) constraints can be modified as part of the overall iterative procedure to improve design. When the objective values of two successive linear programming solutions do not differ from each other by more than a specified amount, the bilinear programming algorithm [10, 11] can be invoked to find an approximate global solution to Problem 2, thereby allowing the optimal design process to continue. Moreover, since the algorithm [10, 11] is a feasible point method, any suboptimal solution can be returned for reformulation of the stick diagram if finding the global optimum is too expensive, especially for large-scale instances of the problem. For large-scale versions of this problem decomposition techniques can be employed to fully exploit the network structure of the constraints in $P$.

By taking a fine enough subpartitioning of $\Omega$, the piecewise linear (convex envelope) underestimating functions can approximate $F$ to within $\varepsilon$ accuracy everywhere over the bounding rectangle $\Omega$. We can derive a worst case complexity bound on the number of partitions required using a modified rectangular subdivision rule. For simplicity, we assume that the underestimating function for each item $\lambda_i x_i y_i$ in $F(x,y)$ is within $\varepsilon/n$ for all $(x_i, y_i) \in \Omega_i$, $i = 1, \ldots, n$. Thus, the underestimating function for $F(x,y)$ is within $\varepsilon$ for all $(x,y) \in \Omega$.

Since convex envelopes are tighter over smaller rectangles, we can partition $\Omega_i$ into $k_i^2$ equal subrectangles by dividing the intervals $[l_i, L_i]$ and $[m_i, M_i]$ into $k_i$ equal subintervals. For simplicity, assume that this is done for each $i$ and that the underestimating function of $F(x,y)$ at $(x,y) \in \Omega$ is given by the sum over $i$ of the convex envelopes of $\lambda_i x_i y_i$, each taken over (any) one of the $k_i^2$ rectangular subsets of $\Omega_i$ which contains $(x_i, y_i)$.

**Theorem.** Under the above hypotheses, the underestimating function for $\lambda_i x_i y_i$ is within $\varepsilon/n$ for every $(x_i, y_i) \in \Omega_i$ if $\Omega_i$ is partitioned into

$$k_i^2 \geq \frac{n|\lambda_i|A_i}{2\varepsilon}$$

subrectangles, where $A_i = (M_i - m_i)(L_i - l_i)$ is the area of rectangle $\Omega_i$.

*Proof.* For brevity, we only sketch the proof here since the result follows immediately from Theorem 11 in [11] and the observation that the maximum difference between the concave envelope (see [11]) and the convex envelope of $\lambda_i x_i y_i$ over $\Omega_i$ occurs at the midpoint $\frac{1}{2}(l_i + L_i, m_i + M_i)^T$ of the rectangle. At this midpoint, the difference between these two functions is $|\lambda_i|A_i/2$. This implies that the convex envelope is within $|\lambda_i|A_i/2$ of $\lambda_i x_i y_i$ for every $(x_i, y_i) \in \Omega_i$. If $\Omega_i$ is partitioned into $k_i^2$ *equal* rectangles by dividing each interval $[l_i, L_i]$ and $[m_i, M_i]$ into $k_i$ equal subintervals, then the convex envelope over any of these rectangles is within $\varepsilon/n$ of $\lambda_i x_i y_i$ if we require

$$\frac{|\lambda_i|A_i}{2k_i^2} \leq \frac{\varepsilon}{n}. \qquad \square$$

**Corollary.** The underestimating function for $F(x,y)$ is within $\varepsilon$ for every $(x,y) \in \Omega$ if each $\Omega_i$ is partitioned into at least $k_i^2 \geq n|\lambda_i|A_i/2\varepsilon$. In this way, $\Omega$ is partioned into $\Pi_{i=1}^{n} k_i^2$ hyperrectangles.

**Remark.** Since fine accuracy is only needed in a neighborhood of a solution, on the average, it is expected that enumerative schemes based on branch and bound will achieve a desired accuracy $\varepsilon$ in a neighborhood of a global solution within a significantly fewer number of partitions.

## 4. CONCLUDING REMARKS

In the last two decades there has been a growing interest in global optimization problems. Many new algorithms and approximate methods have been proposed [7].

Bilinear programming (Section 3) can be viewed as a global optimization problem of a weighted sum of areas. Therefore, it is clear that bilinear programming techniques can be used in the solution of VLSI problems that involve area minimization. In addition to the compaction problem, global optimization can be used in other VLSI problems (see, *e.g.*, [5]).

As mentioned earlier, the purpose of this note is to communicate a new global optimization approach for VLSI. At the moment we have no computational experience. Other approaches to this problem include simulated annealing [12]. A comparison of the proposed approach with the simulated annealing (or other) procedures in current practice will determine the practicality and efficiency of the proposed method.

## ACKNOWLEDGEMENTS

The authors would like to thank the referees for many constructive comments. In particular, one of the referees brought to our attention reference [12].

## REFERENCES

[1] R. K. Brayton, D. G. Hachtel, and A. L. Sangiovanni-Vincentelli, "A Survey of Optimization Techniques for Integrated-Circuit design", *Proceedings of the IEEE*, **69(10)** (1981), p. 1334.

[2] M. J. Ciesieski and E. Kinnen, "An Analytic Method for Compacting Routing Area in Integrated Circuits", *Proceedings of the 19th Design Automation Conference*, 1982, p. 30.

[3] G. Kedem and H. Watanabe, "Optimization Techniques for IC Layout and Compaction", *Proceedings of IEEE International Conference in Computer Design: VLSI in Computers*, 1983, p. 709.

[4] K. Maling, S. H. Mueller and W. R. Heller, "On Finding Most Optimal Rectangular Package Plans", *Proceedings of the 19th Design Automation Conference*, 1982, p. 663.

[5] J. Soukup, "Circuit Layout", *Proceedings of the IEEE*, **69(10)** (1981), p. 1281.

[6] Watanabe, H., "IC Layout Generation and Compaction Using Mathematical Optimization, *Ph.D. Thesis, Computer Science Department, University of Rochester, Rochester, New York*, 1984.

[7] P. M. Pardalos and J. B. Rosen, *Constrained Global Optimization: Algorithms and Applications*. Berlin: Springer-Verlag, 1987.

[8] C. Mead and L. Conway, *Introduction to VLSI Systems*. Reading, Massachusetts: Addison-Wesley, 1980.

[9] S. L. Lin and J. Allen, "Minplex – A Compactor that Minimizes the Bounding Rectangle and Individual Rectangles in a Layout", *Proceedings of the 23rd Design Automation Conference*, 1986, p. 123.

[10] F. A. Al-Khayyal and J. E. Falk, "Jointly Constrained Biconvex Programming", *Mathematics of Operations Research*, **8(2)** (1983), p. 273.

[11] F. A. Al-Khayyal, "Jointly Constrained Bilinear Programs and Related Problems: an Overview, *Computers and Mathematics with Applications*, **19(11)** (1990), p. 53.

[12] Carl Sechen, *VLSI Placement and Global Routing Using Simulated Annealing*, Boston: Kluwer Academic Publishers, (1988).

**Paper Received 27 September 1989; Revised 2 January 1990, 15 July 1990.**