Analytical Placement: A Linear or a Quadratic Objective Function?

GEORG SIGL, KONRAD DOLL, FRANK M. JOHANNES
Institute of Electronic Design Automation, Department of Electrical Engineering
Technical University of Munich, D-8000 Munich 2, Germany

Abstract

This paper addresses the problem of cell placement which is considered crucial for layout quality. Based on the combined analytical and partitioning strategy successfully applied in the GORDIAN placement tool, we discuss the consequences of using linear or quadratic objective functions. By joining the linear objective with an efficient quadratic programming approach, and by applying a refined iterative partitioning scheme, we obtain placements of excellent quality. The effect of a quadratic and a linear objective function on the chip area after final routing is demonstrated for benchmark circuits and other circuits with up to 21 000 cells.

1 Introduction

One of the most challenging problems during VLSI layout synthesis is the placement of the components on the chip. They must be placed in such a way that the chip can be routed efficiently and all timing requirements can be satisfied. This should be accomplished in reasonable computation time even for circuits with tens of thousands of modules. The toughest problem, however, is that all of these tasks must be achieved with wiring models that only partially reflect the actual wiring demands. The choice of the appropriate model and a suitable objective function is therefore crucial for every placement algorithm.

Some algorithms model the circuit as a hypergraph, others replace the hyperedges by cliques. The effect of this modeling on the layout quality depends on the objective function. Objective functions are usually based on wiring length or on wiring density.

Mincut algorithms provide good heuristics for minimizing wiring density and have therefore been frequently used for placement [1, 2, 3]. Another class of placement algorithms minimizes wiring length, sometimes as a linear function of the module coordinates, sometimes as a quadratic function. To minimize a linear objective function, linear programming methods [4, 5] and stochastic optimization techniques [6] have been used. Both methods suffer from excessive computation times. A quadratic objective function, however, allows efficient quadratic programming techniques to be applied [7, 8, 9, 10]. Combined with sparse matrix techniques, they provide very fast and memory efficient algorithms to solve the global placement problem.

Recently, algorithms joining quadratic programming with partitioning [11, 12] have been developed for circuits with tens of thousands of modules. The special advantage of the global optimization and partitioning method GORDIAN [12, 13] is the simultaneous treatment of all modules during all levels of partitioning. The efficiency of that approach has been shown with many industrial examples. Therefore we retain the basic strategy of the original GORDIAN algorithm, i.e. alternating global placement and partitioning steps, in our improved placement algorithm GORDIANL.

The modifications concern the objective function for global placement and the partitioning strategy. The differences between linear and quadratic objective functions and their influence on the wiring are discussed in section 2. In section 3 we show how to optimize a linear objective function with quadratic programming techniques. Using efficient algorithms, which are available for quadratic programming, to minimize the linear objective function yields high quality placements in reasonable computation times. A new iterative partitioning method that avoids partitioning decisions based on insufficient data is introduced in section 4. The two methods GORDIAN and GORDIANL are compared in section 5 by benchmark and other standard cell examples with more than 20 000 modules.

As far as we know, this is the first time that a quadratic and a linear objective function for placement are compared in terms of area after final routing. Thus, this paper will help to answer the question of how the objective function influences routability ([14] p. 109).

2 Comparison of linear and quadratic objective functions

This section will discuss the influence of a quadratic and a linear objective function on the placement. It seems to be impossible to derive theoretical statements which show the superiority of one of these objective functions. Therefore we will demonstrate the difference by exam-

©1991 ACM 0-89791-395-7/91/0006/0427 $1.50

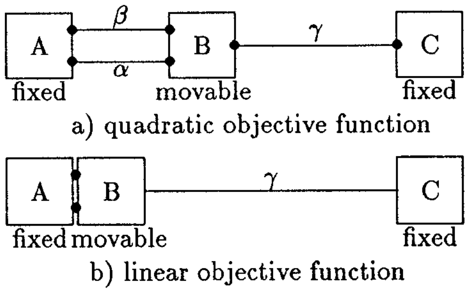

Figure 1: Optimal placements for different objectives

ples. The observations made here will be confirmed in section 5 experimentally.

Figure 1 shows two fixed modules A,C and a movable module B. They are connected by the nets α, β, γ with lengths $l_\alpha, l_\beta, l_\gamma$, respectively. Minimizing the quadratic objective function $\Phi_q = l_\alpha^2 + l_\beta^2 + l_\gamma^2$ yields the placement in fig. 1a with $l_\alpha = l_\beta = \frac{1}{2} l_\gamma$. The minimization of the linear function $\Phi_l = l_\alpha + l_\beta + l_\gamma$ results in the placement in fig. 1b with $l_\alpha = l_\beta = 0$.

It is generally observed that the quadratic objective function tends to make very long nets (net γ in fig. 1) shorter than the linear objective function does, at the expense of the short nets, which become slightly longer. In other words, the standard deviation of the net lengths is smaller for a quadratic objective function than for a linear objective function [14]. How does this influence the wiring?

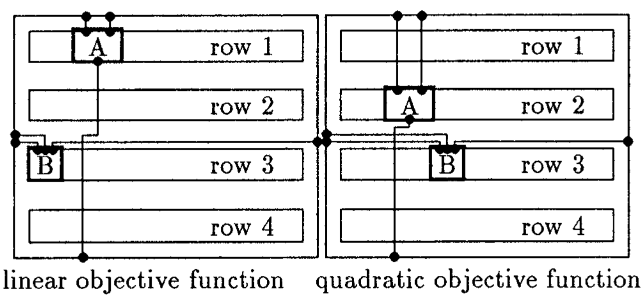

Figure 2: Influence of the objective function

The standard cell circuit in fig. 2 with four rows gives a first answer. Module A (B) is connected with two fixed cells on the top (left) boundary and only one fixed cell on the bottom (right) boundary. The left picture shows the placement obtained with the linear objective function. Module A is placed in row 1 and module B adjacent to the left boundary. The number of feedthroughs generated by the nets connected to module A is three. The routing of the channel between rows 2 and 3 requires at least two tracks. The use of a quadratic objective function leads to the placement in the right picture of fig. 2 with four feedthroughs. At least three tracks are needed to route the channel between row 2 and 3. More tracks as well as more feedthroughs are needed in this placement.

These observations motivated us to take a closer look at the influence of the quadratic and the linear objective function during placement.

3 Global placement with linear objective function

To formulate the objective function for global placement some basic definitions are necessary. The circuit is described by the index sets $\mathcal{M}$ and $\mathcal{N}$ of the modules and the nets, respectively. All modules connected by net ν are in the set $\mathcal{M}_\nu$. Modules and nets are represented by nodes in the graph model of the circuit. The coordinates of the nodes are (x_μ, y_μ) and (x_ν, y_ν), respectively. Fig. 3 illustrates this model.

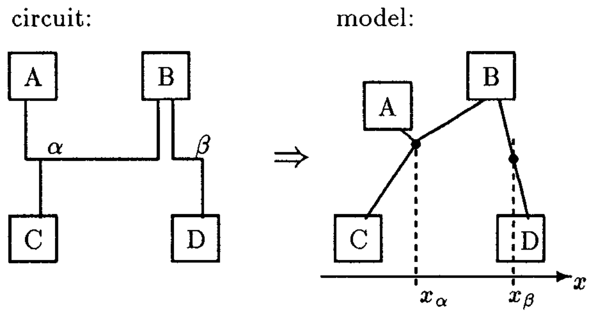

Figure 3: Circuit topology and net model

With these definitions the quadratic objective function Φ_q and the linear objective function Φ_l can be formulated:

$$\Phi_q = \sum_{\nu \in \mathcal{N}} \sum_{\mu \in \mathcal{M}_\nu} (x_{\mu\nu} - x_\nu)^2$$

$$\Phi_l = \sum_{\nu \in \mathcal{N}} \sum_{\mu \in \mathcal{M}_\nu} |x_{\mu\nu} - x_\nu|$$

The pin coordinates $x_{\mu\nu} = x_\mu + \xi_{\mu\nu}$ can be expressed by the module center coordinates x_μ and the relative pin coordinates $\xi_{\mu\nu}$. The coordinates of the net nodes are always in the center of their connected pins, that is $x_\nu = \frac{1}{|\mathcal{M}_\nu|} \sum_{\mu \in \mathcal{M}_\nu} x_\mu$, which is the optimal value for Φ_q. In the two-dimensional case the same objective functions must be formulated for the y-coordinates, but for reasons of brevity they are omitted in this discussion.

The quadratic objective function is used in many analytical placement methods [7, 8, 9, 10, 11, 12]. These methods are also referred to as force-directed placement methods because of the physical analogy with a system of mass points connected by springs. The springs (nets) force the points (modules) to move into positions such that the system has minimum energy.

The main reason for using the quadratic objective function has been that it is continuously differentiable. This means that it can be minimized by solving a linear equation system. Unfortunately, this is not true for the linear objective function Φ_l or other linear objective functions like the half perimeter of the bounding box. Linear objective functions, i.e. the half perimeter, have been minimized by linear programming with a large number of constraints [4, 5]. The experiments, however, show that only medium-sized circuits can be handled in reasonable time. Therefore we prefer methods of quadratic programming to minimize the linear objective function [15].

The linear objective function Φ_l can be rewritten as

$$\Phi_l = \sum_{\nu \in \mathcal{N}} \sum_{\mu \in \mathcal{M}_\nu} \frac{(x_{\mu\nu} - x_\nu)^2}{|x_{\mu\nu} - x_\nu|} = \sum_{\nu \in \mathcal{N}} \sum_{\mu \in \mathcal{M}_\nu} \frac{(x_{\mu\nu} - x_\nu)^2}{g_{\mu\nu}}$$

with $g_{\mu\nu} = |x_{\mu\nu} - x_\nu|$. For constant $g_{\mu\nu}$ the function Φ_l would be quadratic. The only difference between Φ_q and Φ_l is the factor $\frac{1}{g_{\mu\nu}}$, which can be interpreted as a variable spring constant, which increases with decreasing length of the spring.

There is a certain degree of freedom in the choice of the factor. It can be used to fit the placement model (graph model & objective function) better for the actual objective, the area after final routing. Our experiments showed that this can be achieved, if the factor $g_{\mu\nu}$ is replaced by a net specific factor

$$g_\nu = \sum_{\mu \in \mathcal{M}_\nu} |x_{\mu\nu} - x_\nu|.$$

This choice has two advantages. First, the summation reduces the influence of nets with many connected modules and emphasizes the majority of nets connecting only two or three modules. Second, the force on modules close to the net node is reduced. This means for net α in fig. 3 that the force on module A is reduced since $\frac{1}{g_\nu} \ll \frac{1}{g_{\mu\nu}}$. This lower force in the direction to the net node corresponds with the fact that the placement of module A has no influence on the length of the Steiner tree as long as it remains inside the bounding box spanned by the modules B and C.

Nets becoming very short may cause numerical problems during the solution of the global placement. Therefore a lower bound on g_ν has to ensure that g_ν will never be zero. Currently the average module width w_0 is used. An upper bound is not necessary, since the net lengths are bounded by the chip dimension.

procedure *global placement*
$k = 0$;
for each $\nu \in \mathcal{N}$
  $g_\nu^{(k)} = 1$;
endfor
do
  $\Phi_l^{(k)} \rightarrow min$;
  $k = k + 1$;
  for each $\nu \in \mathcal{N}$
    $g_\nu^{(k)} = \max(w_0; \sum_{\mu \in \mathcal{M}_\nu} |x_{\mu\nu} - x_\nu|)$;
  endfor
while $\sum_{\nu \in \mathcal{N}} \left| g_\nu^{(k)} - g_\nu^{(k-1)} \right| > \epsilon$;
endprocedure

Figure 4: Global placement with linear objective

An iterative solution method with iteration count k for the modified objective

$$\Phi_l^{(k)} = \sum_{\nu \in \mathcal{N}} \sum_{\mu \in \mathcal{M}_\nu} \frac{1}{g_\nu^{(k)}} (x_{\mu\nu} - x_\nu)^2$$

is shown in fig. 4. This method alternates quadratic programming for constant $g_\nu^{(k)}$ and updating the factors $g_\nu^{(k)}$. The iteration is terminated when the factors $g_\nu^{(k)}$ no longer change significantly. The quadratic programming problem is solved by a conjugate gradient method with preconditioning by incomplete Cholesky factorization. This method is very well suited for sparse quadratic programming problems.

4 Iterative partitioning

During global placement the objective function is minimized while neglecting geometrical constraints, i.e. overlap of modules. Therefore in [12] global placement is alternated with partitioning steps that generate constraints for the next global placement step. These constraints aim at a better distribution of the modules over the placement area. The set of modules is recursively partitioned into smaller subsets and the placement area is dissected into subregions. The module set $\mathcal{M}_\rho$ of a region ρ of the current dissection is bipartitioned into the subsets $\mathcal{M}_{\rho'}$ and $\mathcal{M}_{\rho''}$ according to the global placement coordinates x_μ such that

$$x_{\mu'} \leq x_{\mu''} \quad \text{for } \mu' \in \mathcal{M}_{\rho'}, \mu'' \in \mathcal{M}_{\rho''}.$$

The sum of the module areas f_μ in both subsets has to be approximately the same, i.e.

$$\sum_{\mu' \in \mathcal{M}_{\rho'}} f_{\mu'} \approx \sum_{\mu'' \in \mathcal{M}_{\rho''}} f_{\mu''}.$$

To distribute the modules better over the whole placement area, positioning constraints fix the center of gravity of modules in set $\mathcal{M}_{\rho'}$ ($\mathcal{M}_{\rho''}$) on the center coordinate $x_{\rho'}$ ($x_{\rho''}$) of the region ρ' (ρ''), i.e.

$$\sum_{\mu' \in \mathcal{M}_{\rho'}} x_{\mu'} f_{\mu'} = x_{\rho'} \sum_{\mu' \in \mathcal{M}_{\rho'}} f_{\mu'}.$$

The next global placement step minimizes the objective function $\Phi_l^{(k)}$ considering these linear equality constraints.

In general, there exist module sets $\mathcal{R} \subseteq \mathcal{M}_\rho$ consisting of modules with equal or nearly equal coordinates. If one of these sets, e.g. $\mathcal{R}^*$, has to be partitioned into sets $\mathcal{R}' \subseteq \mathcal{M}_{\rho'}$ and $\mathcal{R}'' \subseteq \mathcal{M}_{\rho''}$ in order to satisfy the area constraint, the assignment of the modules to the subsets will be arbitrary. With increasing number of modules in the set $\mathcal{R}^*$ the quality of the partitioning decreases, since many assignments are made arbitrarily. Thus, the way the global placement distributes the modules over the placement area has a significant influence on the partitioning step.

The placement with the linear objective function in fig. 7a is obviously much more clustered than the placement with the quadratic objective function. The reason for this clustering of the modules, especially in the center of the chip, is the reduced influence of the few long nets connected to pad cells (cf. sec. 2). These nets, however, are the only nets that force the modules to move away from the center. Therefore we apply a modified partitioning strategy, which forces the modules more and more away from the center of the region.

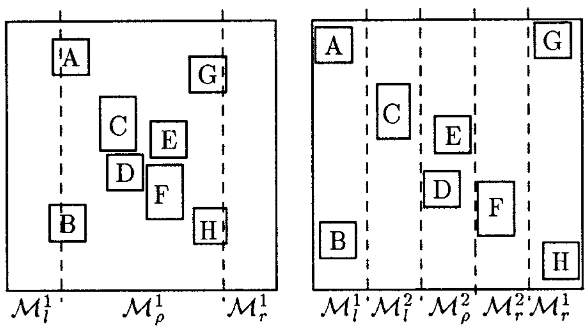

Figure 5: Iterative partitioning

Instead of bipartitioning a region in one step as in [12], it is partitioned iteratively as shown in fig. 5. The left picture shows a typical global placement for a region ρ with the module set $\mathcal{M}_\rho = \{A, B, C, D, E, F, G, H\}$. Some modules are placed far away from the center, i.e. A, B on the left side and G, H on the right side. The other modules are clustered in the center of the region ρ. The first iteration partitions the set $\mathcal{M}_\rho$ into three subsets $\mathcal{M}_l^1 = \{A, B\}, \mathcal{M}_\rho^1 = \{C, D, E, F\}, \mathcal{M}_r^1 = \{G, H\}$ according to the module coordinates x_μ. Thus the partitioning decision is delayed for the modules in $\mathcal{M}_\rho^1$, which are clustered in the center. Positioning constraints force the modules in $\mathcal{M}_l^1$ to move more towards the left and modules in $\mathcal{M}_r^1$ towards the right in the following global placement step. As a direct consequence the modules in the center region $\mathcal{M}_\rho^1$ which are connected to the modules in $\mathcal{M}_l^1$ and $\mathcal{M}_r^1$ will also move away from the center. In a second iterative partitioning step the set $\mathcal{M}_\rho^1$ will be divided into the sets $\mathcal{M}_l^2 = \{C\}, \mathcal{M}_\rho^2 = \{E, D\}, \mathcal{M}_r^2 = \{F\}$. The iterative process will be finished when the set $\mathcal{M}_\rho^i$ becomes empty. The number of modules assigned to the sets $\mathcal{M}_l^i$ and $\mathcal{M}_r^i$ is determined by the area constraint

$$\sum_{\mu \in \mathcal{M}_l^i} f_\mu \approx \sum_{\mu \in \mathcal{M}_r^i} f_\mu \leq \delta \cdot \sum_{\mu \in \mathcal{M}_\rho} f_\mu \quad , \ 0 < \delta \leq 0.5$$

Finally, the bipartitioning of the set $\mathcal{M}_\rho$ is obtained by $\mathcal{M}_{\rho'} = \bigcup \mathcal{M}_l^i$ and $\mathcal{M}_{\rho''} = \bigcup \mathcal{M}_r^i$.

The partitioning process generates a placement with small module overlaps which are eliminated by local moves [3, 16].

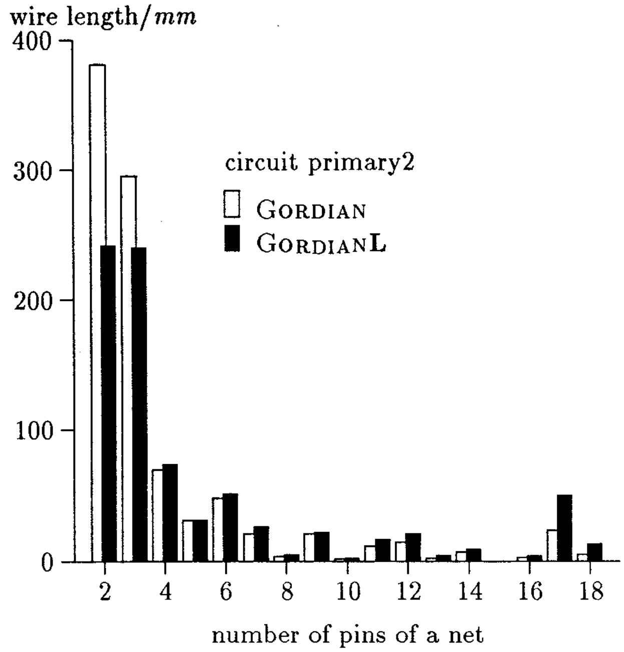

Figure 6: Sum of wire lengths versus #pins

5 Results

Results for benchmark circuits [17, 18] and other standard cell circuits are presented in table 1. The given areas after final routing have been obtained with the VPNR routing package [19]. The areas do not include the left and right boundary channels, since these channels are not routed automatically by the VPNR system. The linear objective function used in GORDIANL yields results with up to 20% less area than the quadratic objective function of the original GORDIAN procedure. Even with the moderate increase in cpu-time from the use of the linear objective function, a circuit with as many as 21 000 modules has been placed in 6 hours (on a VAX 8650). For circuit AVQ, the very time consuming global and final routing steps have been completed only for the placement produced by GORDIANL.

The results of the placement algorithms have also been compared with the VPNR cplrt (combined place and route) algorithm [3]. The bold numbers indicate better results for GORDIANL in almost all cases. These results have been computed about 3 times faster than with VPNR cplrt. VPNR cplrt failed for the large circuit AVQ due to excessive memory requirements.

The reason for the substantial improvements with GORDIANL is the reduction of the net length for nets connecting only 2 and 3 modules. The graph in fig. 6 shows the sum of the net lengths (estimated by the half perimeter of the enclosing rectangle) versus the number of pins of the nets. The contribution of nets with 2 and 3 pins to the total net length is reduced using the linear objective function. The increase of the net lengths of the few nets with more than 3 pins has only a minor effect.

Fig. 7 demonstrates the differences between GOR-

circuit	#modules	#nets	GORDIAN		GORDIANL		VPNR (cplrt)	
			area	cpu	area	cpu	area	cpu
primary1	752	904	23.4	40	22.7	203	**21.8**	767
struct	1888	1920	9.2	113	**6.7**	435	7.1	788
primary2	2907	3028	97.3	260	**85.5**	1180	90.1	2559
biomed	6417	5742	62.2	804	**50.5**	5814	52.7	13825
c1355	554	595	2.8	20	**2.2**	98	2.3	120
c5313	2330	2508	18.1	162	**15.4**	764	17.2	1440
s9234	5597	5844	46.6	250	38.0	1918	**37.7**	5575
20x20	6419	6464	16.5	839	**15.1**	3276	16.5	12791
AVQ	21046	21316		5067	86.9	22989		

Table 1: Results (area in mm^2, cpu-time in *seconds* on VAX 8650)

DIAN and GORDIANL, showing the stepwise placement refinement for the circuit c1355 with the linear and the quadratic objective functions. Wiring length and density can be estimated from the Manhattan minimum spanning trees. The use of the linear objective function results in a 'rat's nest' of lower density and about 30% reduced minimum spanning tree length. The area after final routing could be reduced by 20% as shown in table 1.

6 Conclusions

The choice of the objective function is crucial to an analytical placement method. A linear objective function seems to reflect the actual wiring demands more accurately than the quadratic objective function. This is confirmed by several benchmark circuits and other standard cell circuits. We observe a significant synergetic effect when the linear objective function for global placement is combined with a refined partitioning strategy. The new method GORDIANL yields area improvements of up to 20% after final routing. The main reason for these distinct improvements was the length reduction of nets connecting only two and three pins. Quadratic programming techniques, minimizing the linear objective function, solve the placement problem efficiently, even for circuits with more than 20 000 modules.

Acknowledgements

The authors would like to acknowledge Professor K. Antreich for his support and J. Kleinhans for his valuable suggestions. Thanks also to F. Brglez and K. Kozminski from MCNC for providing us with the VPNR-package and with their results obtained for the benchmark circuits.

References

[1] U. Lauther, "A min-cut placement algorithm for general cell assemblies based on a graph representation," *DAC*, pp. 1–10, 1979.

[2] A. E. Dunlop and B. W. Kernighan, "A procedure for placement of standard-cell VLSI circuits," *IEEE Trans. on CAD*, pp. 92–98, 1985.

[3] P. R. Suaris and G. Kedem, "An algorithm for quadrisection and its application to standard cell placement," *IEEE Trans. on CAS*, pp. 294–303, 1988.

[4] B. X. Weis and D. A. Mlynski, "A graphtheoretic approach to the relative placement problem," *IEEE Trans. on CAS*, pp. 286–293, 1988.

[5] M. B. Jackson and E. S. Kuh, "Performance-driven placement of cell based IC's," *DAC*, pp. 370–375, 1989.

[6] C. Sechen and A. Sangiovanni-Vincentelli, "The Timber Wolf placement and routing package," *Journal of Solid-State Circuits*, pp. 510–522, 1985.

[7] K. M. Hall, "An r-dimensional quadratic placement algorithm," *Management Science*, pp. 219–229, 1970.

[8] K. J. Antreich, F. M. Johannes, and F. H. Kirsch, "A new approach for solving the placement problem using force models," *ISCAS*, pp. 481–486, 1982.

[9] C.-K. Cheng and E. S. Kuh, "Module placement based on resistive network optimization," *IEEE Trans. on CAD*, pp. 218–225, 1984.

[10] J. Frankle and R. M. Karp, "Circuit placements and cost bounds by eigenvector decomposition," *ICCAD*, pp. 414–417, 1986.

[11] R. Tsay, E. S. Kuh, and C. Hsu, "PROUD: A fast sea-of-gates placement algorithm," *DAC*, pp. 318–323, 1988.

[12] J. M. Kleinhans, G. Sigl, and F. M. Johannes, "GORDIAN: A new global optimization / rectangle dissection method for cell placement," *ICCAD*, pp. 506–509, 1988.

[13] J. M. Kleinhans, G. Sigl, F. M. Johannes, and K. J. Antreich, "GORDIAN: VLSI placement by quadratic programming and slicing optimization," *IEEE Trans. on CAD*, to be published 1991.

[14] B. T. Preas and M. J. Lorenzetti, *Physical Design Automation of VLSI Systems*. Benjamin/Cummings Publishing Company, 1988.

[15] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*. Academic Press, Inc., 1981.

[16] G. Sigl and U. Schlichtmann, "Goal oriented slicing enumeration through shape function clipping," *EDAC*, pp. 361–365, 1991.

[17] International Workshop on Layout Synthesis, Microelectronics Center of North Carolina, 1990.

[18] F. Brglez and H. Fujiwara, "A neutral netlist of 10 combinational benchmark circuits and a target translator in Fortran," *ISCAS*, 1985.

[19] VPNR Users Guide, MCNC Technical Report, Microelectronics Center of North Carolina, 1988.

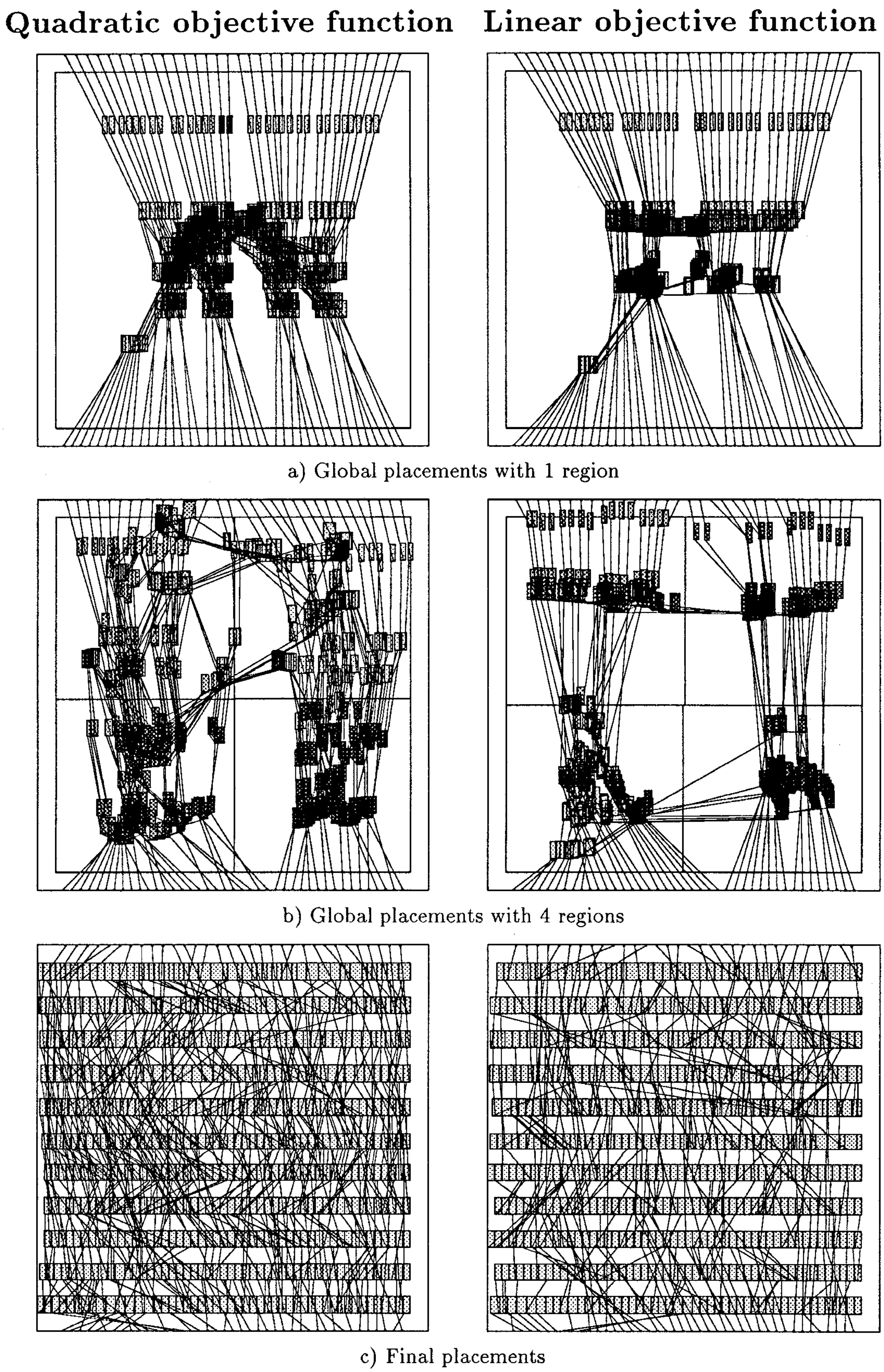

Figure 7: Placement refinement with quadratic and linear objective function