

Influence of Professor T. C. Hu's Works on Fundamental Approaches in Layout

Andrew B. Kahng

UCSD Departments of CSE and ECE, La Jolla, CA 92093 USA
abk@ucsd.edu

ABSTRACT

Professor T. C. Hu has made numerous pioneering and fundamental contributions in combinatorial algorithms, mathematical programming and operations research. His seminal 1985 IEEE book *VLSI Circuit Layout: Theory and Design*, coedited with Prof. E. S. Kuh, shaped algorithmic and optimization perspectives, as well as basic frameworks, for IC physical design throughout the following decades [44]. Indeed, Professor Hu gave a keynote address at the very first International Symposium on Physical Design, in April 1997 [41]. His research approach of (i) studying small and/or extremal cases first, (ii) always seeking to establish error bounds or other properties of heuristic outcomes (hence, always seeking to understand optimal solutions), (iii) approaching new problems from as fresh a perspective as possible, and (iv) pursuing simplicity and beauty in both formulation and exposition, has influenced generations of researchers. This paper complements [20] in recounting highlights of Professor Hu's contributions to, and impacts on, physical design. The focus is on several problems of "connection", as well as the concept of "shadow price", as they relate to layout.

ACM Reference Format:

Andrew B. Kahng. 2018. Influence of Professor T. C. Hu's Works on Fundamental Approaches in Layout. In *ISPD'18: 2018 International Symposium on Physical Design, March 25–28, 2018, Monterey, CA, USA*. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3177540.3177563>

1 INTRODUCTION

VLSI layout encompasses floorplanning, placement and routing, and is at the heart of integrated-circuit physical design. The pursuit of the "science" of VLSI layout has provided an important nexus of mathematics, graph theory, computer science, combinatorial algorithms, electrical engineering, device physics and optimization. Layout is where weighted hypergraphs and grid-graphs meet modules, signal nets, design rules, and a host of variant objectives and constraints that evolve continuously with semiconductor technology.

Professor T. C. Hu's works on VLSI layout, as well as his 1985 co-edited book *VLSI Circuit Layout: Theory and Design* [44], have made a number of highly influential contributions to the field of physical design. Starting over 30 years ago, his works brought combinatorial

optimization and mathematical programming formulations and methods that had not been previously applied to layout problems. As reviewed in [20], numerous works of Professor Hu study the tree representation of inherent structure in VLSI circuits, as well as applications of duality (flows and cuts; shadow price). To this day, [44] remains a key reference work for hypergraph and graph models that underlie VLSI layout formulations, and for fundamental approaches to routing.

Professor Hu's papers and books consistently reflect his unique ability to combine geometric, graph-theoretic and combinatorial-algorithmic concepts. His novel framing of VLSI layout problems can be seen in his keynote address [41] at the very first ISPD in 1997, and in his 1982 book *Combinatorial Algorithms* [40], extended with M. T. Shing in 2002 [47]. The latter presents unifying treatments of (i) the minimum spanning tree and shortest-paths tree constructions, (ii) the discovery of underlying cut structure in networks without maximum flow [60] [63], and (iii) the unification of breadth- and depth-first traversals of a given graph. These ideas are visible throughout the IC physical design literature, e.g., in (i) the Prim-Dijkstra tradeoff of [8], (ii) the ESC clustering method of [24], and (iii) the "window" method [11].

In the following, Section 2 reviews Professor Hu's concept of "TACP" (Tentative Assignment, Competitive Pricing), whose application can be seen particularly in the floorplanning and global placement literatures. Section 3 examines the 1973 result of Adolphson and Hu, which was the first to propose a cut-based placement approach (in the context of linear placement), and which moreover established early bounds on achievable wirelength minimization based on the cut structure of a given circuit. Section 4 discusses the 1985 Hu-Moerder hyperedge net model, its original motivation of capturing flow properties of a circuit hypergraph, and its modern use in analytic placement. Section 5 concludes with a review of the 1993 Prim-Dijkstra routing tree construction, which has been widely used in industry IC design tools for over 20 years. Section 6 discusses the problem of finding a "wide path", and a network flow-based method that is an outgrowth from a 1992 result on the Plateau's minimum surface problem. These works share common themes of duality and "connection", in addition to their application to fundamental layout formulations of floorplanning, placement and routing.

2 "TACP" AND SHADOW PRICE

The "classical" floorplanning formulation studied in much of the academic literature seeks to shape and pack all blocks, such that no blocks overlap and the enclosing layout region has minimum area while satisfying aspect ratio constraints. By contrast, *fixed-outline floorplanning* (FOFP) [49] is motivated by the modern fixed-die (as opposed to variable-die) layout context: the aspect ratio of the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ISPD'18, March 25–28, 2018, Monterey, CA, USA

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-5626-8/18/03...\$15.00

<https://doi.org/10.1145/3177540.3177563>

floorplan is fixed, but the aspect ratios and indeed the shapes of the blocks can vary. At ISPD-2000, [49] proposed a *perfect rectilinear floorplanning* formulation that seeks zero-whitespace, perfectly packed rectilinear floorplans in a fixed-die regime. In the FOFP regime, floorplanning regains its proper focus on *connectivity*, timing and performance; packing itself becomes a non-issue.

FOFP provides a continuum between floorplanning and coarse placement in its support of global interconnects and performance optimizations. To avoid overconstraining, there is no restriction on block shapes. It is even possible for blocks to overlap, as long as there is no violation of bounds on the cell area (i.e., contents of given blocks) that is assigned to any region of the layout. This concept was originated by Professor Hu in the 1980s. Footnote 9 of [49] writes, “For example, two blocks with equal amounts of cell area could be placed into adjacent disjoint regions, with each block having depth = 1 in its respective region. An alternative would be to place each block with uniform depth = 1/2 into the union of the two regions. This idea was first proposed in 1987 by Prof. T. C. Hu in the context of a “TACP” (tentative assignment and competitive pricing) approach to placement.” Figure 1 illustrates the above idea.

Based on the FOFP formulation, Adya et al. [2] suggest new objective functions to drive simulated annealing and new types of moves that better guide local search in hierarchical design to improve wirelength and aspect ratios of blocks. Liu et al. [57] propose an algorithm that uses sequence-pair representation and instance augmentation to optimize the floorplan. Lin et al. [54] develop an evolutionary search method to minimize the area. More recent works of [22][58][66] propose several other FOFP optimizations, including a two-stage convex optimization methodology, insertion-after-remove (IAR) technique, and deferred decision making. These methods improve runtime, wirelength and area over previous approaches.

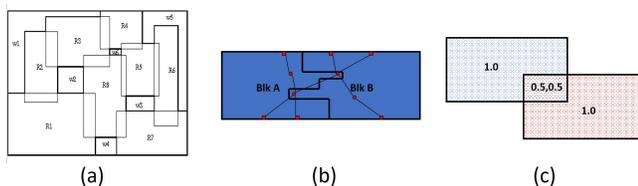


Figure 1: Fixed-Outline Floorplanning: (a) coarse placement-like global floorplanning; (b) irregular block shapes; and (c) example of tentative assignment.

The concept of competitive pricing in a TACP iteration is based on the *shadow price* in linear programming duality. As discussed in [20], column generation and shadow price have been applied to VLSI global routing, with primal-dual iterations being widely studied approximately 20 years ago [18] [4] [5]. Shadow price has subsequently been used in global placement, which lies on the placement continuum with FOFP. Equation (1) shows the classical analytical global placement objective, where $W(\mathbf{v})$ is the approximated total half-perimeter wirelength, and $D(\mathbf{v})$ represents the global density cost. These two terms are connected using a global Lagrangian multiplier λ .

The authors of the recent RePlace global placement framework [21] propose a new constraint-oriented local-density function that incorporates a constraint-oriented local-density penalty factor for each placement bin i , as shown in Equation (2). Compared to the global density penalty factor which balances wirelength and cell spreading using only once coefficient, the new local-density function comprehends local density overflow at per-bin granularity, thus accelerating cell spreading for overflowed bins to resolve cell overlapping, while preserving the wirelength elsewhere. The concept of iterative pricing and (spatial location) assignment will no doubt continue to find further applications in physical design, e.g., co-optimization of power delivery network and placement. Further, the “fixed-outline floorplanning” problem of determining block shapes, utilizations, pin locations, etc. in an SOC floorplan remains a critical challenge for IC design teams.

$$\min_{\mathbf{v}} f(\mathbf{v}) = W(\mathbf{v}) + \lambda D(\mathbf{v}) \quad (1)$$

$$\min_{\mathbf{v}} f(\mathbf{v}) = W(\mathbf{v}) + \sum_i \lambda_i D_i(\mathbf{v}) \quad (2)$$

3 LINEAR PLACEMENT

Placement is a fundamental problem in many applications. The module placement problem, originating from Steinberg’s backboard wiring problem, dates from 1961 and is one of the classical problems of VLSI layout. In this formulation, a set of n pins connected by wires should be placed into n holes, one pin per hole, such that total cost (wirelength) is minimized. A special case of the problem is the *optimal linear ordering* (O. L. O.) problem, where all holes are in a line, and are unit distance apart. The problem can be described using a graph, where each node represents a pin, and each edge represent a wire between pins. Each edge can be associated with a capacity representing the number of connections between its two nodes. The work of Adolphson and Hu [1] provided fundamental insights and bounds for the linear ordering problem, based on theory of maximum flows and minimum cuts in networks.

Gomory and Hu [31] show that $n(n-1)/2$ max flow values between any two source, sink nodes can be obtained with only $n-1$ max flow problems, giving $n-1$ “fundamental cuts”. Adolphson and Hu [1] show that the sum of the values of the $n-1$ fundamental cuts is a lower bound on the total wirelength of the O.L.O. problem, and is the solution to the O.L.O. problem if the Gomory-Hu cut tree is a chain. The above is stated by Cheng [19] as:

Theorem 1. The min-cut defines an ordered partition that is consistent with an optimal vertex order in the linear placement problem.

Adolphson and Hu [1] also study the O.L.O. for *two- k -chain* instances, and propose an efficient $O(n \log n)$ algorithm in the special case where the graph is a rooted tree. A rooted tree can also represent a job sequencing problem where each non-root node i represents a job, and has an associated linear delay cost $V(i)$ as well as a time $T(i)$ needed to finish the job. The objective of job sequencing is to find the job execution sequence ρ that minimizes the sum of linear delay costs when only one machine executes all jobs and all precedence constraints in the tree are satisfied. More precisely, the goal is to minimize the cost $c(\rho) = \sum_i V(i) \sum \{T(j) : \rho(j) \leq \rho(i)\}$. Adolphson and Hu prove equivalence to the job sequencing problem with tree-like precedence relationships and linear delay costs, as well as the method of Horn [35]. Based on the work of Adolphson

and Hu [1], Cheng in 1987 [19] proposed a recursive partitioning algorithm for arbitrary graphs which generates provably optimal placement solutions.

Minimum Cuts (Maximum Flows) in Placement. The early works of [1] [19] presage the universal application of the recursive min-cut approach to VLSI placement within commercial tools up through at least the mid-2000s. (In later years, min-cut has been often applied in conjunction with analytic methods.) Capo [16] uses a top-down, min-cut bisection algorithm, seeking to decompose a given placement instance into smaller instances by subdividing the placement region, assigning modules to subregions and inducing corresponding netlist sub-hypergraphs. Multiple types of min-cut partitioner are used that facilitate partitioning and improve efficiency; terminal propagation and whitespace management methods are also enabling to Capo solution quality. Numerous other works such as [3] also develop general-purpose global placers – applicable to both standard cell-based and mixed-size contexts – based on the min-cut paradigm.

The duality between maximum flows and minimum cuts is explicitly applied by Yang and Wong [67], who approach finding a balanced minimum cut in a netlist hypergraph from a maximum-flow perspective. Their work models the circuit netlist by a flow network, and heuristically achieves a balanced bipartition using repeated max-flow min-cut computations. Works such as [17] examine the time-quality tradeoff continuum from multilevel KL-FM partitioning, to flat KL-FM partitioning, to flow-based approaches, to implicit enumeration (branch-and-bound) within the top-down placement framework. Since the number of cell rows in a standard-cell block is often smaller than the number of cells per row, the “one-dimensional” linear placement problem is of special interest even within top-down placement of a “two-dimensional” layout.

Linear Placement Today. Since [1], there have been many linear placement-related problem formulations and results in the physical design literature. Indeed, today there is a tremendous renewal of attention to the linear placement problem. This is a consequence of increasingly intrusive manufacturing-induced placement constraints (minimum implant area, avoidance of OD notches, etc.) as well as heightened interference between the placement problem and various details of power delivery and pin access. Another driver for attention to linear placement is that it is an obvious opportunity for “end-case” recovery of layout quality, as noted in [17].

The work of [50] studies a type of linear placement problem in detailed placement. The authors propose a *single-row placement problem* which differs from classical linear placement in three ways: (i) cells (modules) have variable width; (ii) each row has fixed length with free sites; and (iii) cell ordering is fixed. The objective is to legalize all cells while minimizing the wirelength, given that all cells in other rows are fixed, as shown in Figure 2. In the figure, C_1, \dots, C_n gives the fixed left-to-right order of movable cells. For each legal position s_j of each movable cell C_i , since the optimal ordering is given and fixed, a minimum-cost constrained prefix placement C_1, C_2, \dots, C_i , subject to the position of C_i being at or to the left of s_j , can be obtained. [50] gives a dynamic programming technique with time complexity $O(m^2)$ where m is the number of nets incident to cells in the given row. The dynamic programming is applied by using the prefix placement solution $P_{i-1}(s_j)$ to compute

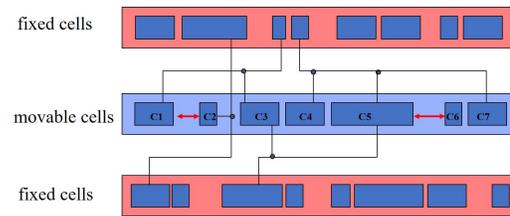


Figure 2: Ordered single-row placement.

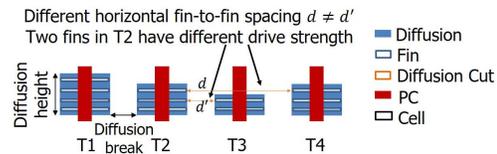


Figure 3: Neighbor diffusion effect caused by diffusion steps.

$P_i(s_j)$. A $O(m \log m)$ ‘clumping’ technique further exploits the convexity of the wirelength objective. Iterative improvement of cell ordering within a row can be run before applying this methodology.

Recent developments in linear placement reach beyond the ordered single-row placement, to multi-row detailed placement considering layout-dependent effects [27][32][55][56]. The work of [32] considers the neighbor diffusion effect caused by diffusion steps between adjacent standard cells, as shown in Figure 3. An optimal single-row dynamic programming-based approach is proposed to minimize the inter-cell diffusion cost and support cell variants, relocating and reordering. The authors further extend to an optimal double-row detailed placement supporting movable double-height cells.

One might speculate that “disconnects” between placer turnaround requirements and the runtime complexity of maximum flow or minimum cut calculation, as well as between the linear ordering problem and the two-dimensional nature of layout, led the physical design field to bypass the use of flow network “structure” in VLSI placement. Yet, the clear trend in recent years has been toward mathematical programming and combinatorial-algorithmic approaches that pay high runtime complexity as the price of solving real-world objectives and constraints. Looking forward, known mappings between one- and two-dimensional problem embeddings, as in [10], may help reconnect linear placement to the global placement and floorplanning problems. It may also be possible for flow- and cut-based frameworks to rejoin the toolkit for placement optimization in the future.

4 NET MODELING

In their 1985 paper, Hu and Moerder [45] propose a new hyperedge *net model* which use p pin nodes and one *star* node, to represent a p -pin hyperedge. A given netlist hypergraph is transformed by adding one star node for each signal net, and connecting the star node via a graph edge to each of the net’s pin nodes. In the Hu-Moerder net model, we again see the motivation of network flows: specifically, it enables extension of network flow techniques to placement of modules on a chip. Figure 4(b) shows the Hu-Moerder hypergraph model of the set of modules and nets shown in Figure 4(a). The

authors of [45] show that the max-flow min-cut theorem [53] can be used to find a minimum-cut bipartitioning of a hypergraph. They further describe an algorithm to construct a tree that is flow-equivalent to a given hypergraph.¹

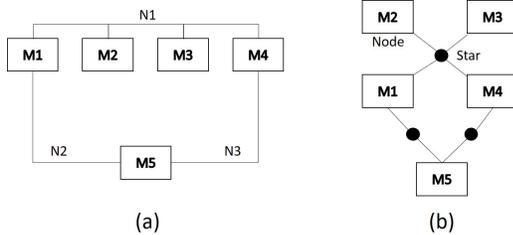


Figure 4: (a) Example circuit with 5 modules and 3 nets. (b) After transformation using the Hu-Moerder hyperedge net model.

The Hu-Moerder hyperedge model gives a very early, elegant answer to a question that challenges the VLSI layout community even today. Namely, how should a hyperedge of a hypergraph be *fairly* modeled by graph edges in a graph model of the hypergraph? This question is crucial to the application of analytic placement and to the ability to exploit sparse-matrix codes for layout applications.

Many works have proposed net models to represent hypergraphs as graphs. Weighted clique ($C(p, 2)$ edges), directed star ($p - 1$ edges from the source to sink pins), spanning tree, etc. models have all been considered, but have respective disadvantages and advantages according to considerations of sparsity, whether the context is placement (based on half-perimeter wirelength) or partitioning (based on net cut), etc. A number of noted works on analytic placement, such as PROUD [65], Gordian [51] and BonnPlace [14], discuss the challenges of applying hyperedge net models in recursive placement and/or placement-based partitioning. Ihler et al. [48] proved in 1993 that it is impossible to achieve a net model that fairly represents the net cut properties of a hyperedge with more than three nodes. The star net model itself has been separately realized and used over the years, e.g., Brenner and Vygen describe a star net model in [15] and subsequently apply it in BonnPlace global placement [14].

Since its publication 33 years ago, the Hu-Moerder model arguably has not received the attention that it deserves. The model has important, useful qualities: it is sparse and symmetric, and it enables exact representation of net cut cost. It may be that in an era before 64-bit addressing, and before the emergence of sophisticated memory pool management and containers, the notion of rewriting the netlist to have essentially double the original number of vertices may have been unappealing. However, with today's ubiquitous tight loops involving buffer/inverter insertion, fanout clustering, and other on-the-fly netlist topology changes during physical optimization, it may be that past obstacles to the use of the star net model no longer exist.

5 PRIM-DIJKSTRA

The 1993 Prim-Dijkstra (PD) algorithm [8] has been used extensively in industry for construction of spanning trees with good

¹This cut-tree is similar to the Gomory-Hu cut-tree and result on multi-terminal maximum flows [31] that is reviewed in [20]. Professor Hu's work [37] subsequently applies the Gomory-Hu cut-tree in the context of multicommodity flows.

balance between tree wirelength (WL) and source-to-sink path-lengths (PLs). Leading electronic design automation (EDA) tools and semiconductor company methodologies have used the *PD* algorithm extensively in their design flows, as evidenced by a number of patents assigned to IBM, Synopsys, Cadence, etc. [33] [13] [34] [29] [30] [62] [9] [7].

The *PD* algorithm merges the Prim and Dijkstra spanning tree constructions [61][26] to explicitly trade off tree WL (lightness) and source-to-sink PLs (shallowness). In their paper, the authors of [8] describe two versions of the algorithm, *PD1* and *PD2*.

Starting with just the source node v_0 , *PD1* iteratively adds the edge e_{ij} and sink v_j to the tree such that $d_{ij} + \alpha \cdot l_i$ is minimized, where node v_i is in the current tree and v_j is not in the current tree. Here, d_{ij} is the distance between nodes v_i and v_j ; l_i is the pathlength from the source v_0 to v_i in the current tree; and α is a weighting factor with $0 \leq \alpha \leq 1$. When $\alpha = 0$, *PD1* constructs Prim's minimum spanning tree (MST) [61], and with $\alpha = 1$, *PD1* constructs a shortest-path tree (SPT), identically to Dijkstra's algorithm [26]. As α increases from 0 to 1, *PD1* constructs trees with larger WLs and shorter PLs.

The *PD2* construction gives a second Prim-Dijkstra tradeoff, by iteratively adding the edge e_{ij} and sink v_j to the tree such that $(l_i^p + d_{ij})^{1/p}$ is minimized. Here, p is a parameter whose value satisfies $1 \leq p < \infty$. When $p = \infty$, *PD2* produces a tree identical to Prim's MST, and when $p = 1$, *PD2* yields an SPT. Figure 5 shows *PD1* and *PD2* trees obtained with various parameter values for a 9-terminal net.

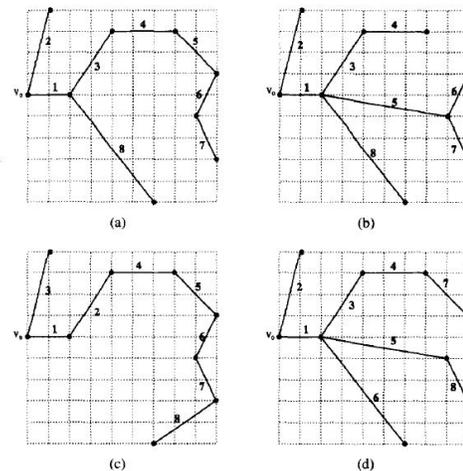


Figure 5: Tree constructions with *PD1* and *PD2* for a 9-terminal net. The edge labels give the order in which the algorithms add the edges into the tree. *PD1* constructions with $\alpha = 1/3$ and $\alpha = 2/3$ are shown in (a) and (b), respectively. *PD2* constructions with $p = 3$ and $p = 3/2$ are shown in (c) and (d), respectively.

Prim's and Dijkstra's algorithms are well-known, fundamental graph algorithms that are basic to undergraduate discrete math and computer science curricula. The *PD* heuristic that blends these two greedy algorithms could very well be placed beside them in the same textbook. *PD* is very effective in practice, but simple enough to be implemented as homework in an algorithms course. The

approach has also found broad appeal outside of IC design, with myriad applications including flood control [25], biomedical [68], military [59], wireless sensor networks [64], etc. At the same time, discovery and exploitation of the *PD* tradeoff took place 30+ years after the Prim and Dijkstra algorithms were published, suggesting non-obviousness and non-triviality.

It should be noted that the *PD* algorithm's construction of a *lem* spanning tree, as opposed to a *Steiner tree*, is advantageous in today's physical design tool chain. Spanning trees are widely used for global routing, since they provide an obvious way for the global router to decompose multi-fanout nets into two-pin nets. Even though rectilinear Steiner trees are required for actual realization of interconnect wires, Steiner trees are not preferred during the early routing stages because: (i) constructing Steiner trees is more time-consuming due to the added complexity of handling Steiner points, and (ii) Steiner points become unnecessary constraints that restrict the freedom of the global router to resolve congestion. Hence, spanning trees are typically preferred to Steiner trees.

Since the publication of [8], a number of researchers have continued to explore the tradeoff between lightness and shallowness of interconnection trees. The work of [28] achieves optimal (up to constant factors) tradeoffs of tree depth, tree weight, maximum degree and shallowness in a "narrow-shallow-low-light" construction. The merits of the *PD spanning tree* construction have also been revisited over the years. These merits are reasserted in the DAC-2006 work [12], whose authors argue that the *PD WL* is sufficiently close to minimal that it is practically 'free'. Their work concludes that *PD* obtains the best tradeoff between WL and PL compared to other spanning tree constructions such as BRBC [23], KRY [52], etc.

Finally, the scaling of power, performance, and area density has been exceptionally challenging in recent process nodes. This has brought renewed attention to the challenge of minimizing routing cost while optimizing delay or skew metrics. Indeed, for today's designs that are highly power-sensitive, even a 1% reduction in power provides considerable benefits. Consequently, even a small WL savings can have a high impact on value. A recent work [6] follows this motivation and proposes a new *PD-II* construction which directly improves upon the original *PD* construction by repairing the tree to simultaneously reduce both WL and PL. It seems inevitable that further research will be needed to similarly improve required arrival time, skew, prescribed-delay, per-sink radius, and other tradeoffs with tree cost that arise in physical design.

6 FINDING A WIDE PATH

Professor Hu has also made early contributions to connection-finding, which is a basic element of any routing approach. For example, the α - β routing of [46] finds connections when there exist both edge and vertex costs along a routing path. Notably, when the cost of traversing a vertex depends on whether a turn is being made at that vertex, this induces a "history-dependence" left unaddressed in previous works. The method of [46] unifies elements of Dijkstra's algorithm and best-first (A^*) search.

The 1993 work [43] studies the problem of *robust path finding* (e.g., for a mobile agent in a general environment), which seeks a minimum-cost source-destination path *having prescribed width*. Figure 6 gives a cartoon of a source-destination (i.e., s - t) path with

prescribed width d , as might be required by a robot that has finite width.

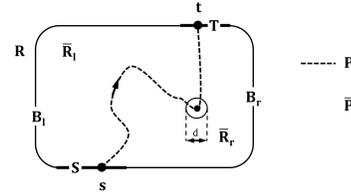


Figure 6: A d -separating path \bar{P} of width d between two points $s \in S$ and $t \in T$ of the boundary of a region R . When the path has width d , every point of \bar{R}_l is separated from every point of \bar{R}_r by a distance of at least d .

Hu et al. [43] exploit the duality between cuts and flows to find a minimum-cost *robust* (i.e., a wide path) in a routing environment between given source and terminal nodes, s and t . A flow network is constructed based on a discretized (grid-graph) representation of the routing environment. By construction, a minimum cut between two appropriately chosen other vertices in this network will contain all vertices and edges inside a (minimum-cost) robust path between s and t . (A similar discretization is used in [42] to solve the discrete version of Plateau's problem of finding a minimum surface with given boundary.)

So that a maximum-flow computation will return a robust path, Hu et al. [43] superimpose a mesh network topology over region R . After assigning node weight to each node in the network, they connect each node to every neighboring node that is within distance d . Finally, they connect the source s' and sink t' to part of the boundary of region R such that a minimum $s'-t'$ cut corresponds to a wide path between s and t . Figure 7 illustrates the transformation from motion planning instance to network flow instance.

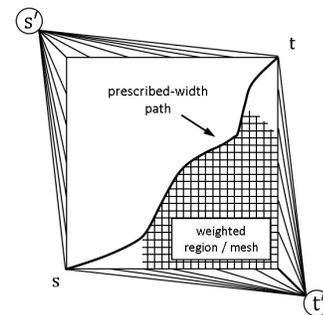


Figure 7: A robust motion planning instance transformed into a network flow instance.

Today, *wide paths* are relevant to many difficult and time-consuming connection formulations, such as top-level bus routing, bus feedthrough determination, etc. Also, as illustrated in Figure 8, IC package routing problems usually use traces of various width for different nets, depending on respective power integrity and signal integrity requirements. Most existing commercial package routers do not support such freedom of trace width. However, in a sequential routing scheme, *wide path* finding can provide routes with specified width in order to meet package power and signal integrity requirements.

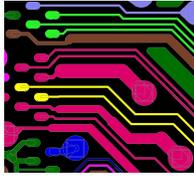


Figure 8: Clip from chip package routing showing traces of various widths.

7 CONCLUSIONS

This paper has reviewed several fundamental contributions of Professor T. C. Hu to the field of VLSI layout, focusing on topics of connection and duality that are at the heart of modern floorplanning, placement and routing. The works reviewed here provide very early formulations and algorithmic solutions based on mathematical programming and network flows. As today's layout problems become more complex and challenging, the combinatorial frameworks and approaches proposed by Professor Hu become more compelling, and may be enabling to future physical design innovations.

8 ACKNOWLEDGMENTS

Above all, many thanks are due to Professor Hu for his guidance, inspiration, patience and help that has been given so generously to students and colleagues. I also thank Sriram Venkatesh, Lutong Wang and Bangqi Xu for their help in compiling and drafting the above material.

REFERENCES

- [1] D. Adolphson and T. C. Hu, "Optimal Linear Ordering", *SIAM J. Appl. Math.* 25(3) (1973), pp. 403-423.
- [2] S. N. Adya and I. L. Markov, "Fixed-Outline Floorplanning: Enabling Hierarchical Design", *IEEE TVLSI* 11(6) (2003), pp. 1120-1135.
- [3] A. R. Agnihotri, S. Ono and P. H. Madden, "Recursive Bisection Placement: Feng Shui 5.0 Implementation Details", *Proc. ISPD*, 2005, pp. 230-232.
- [4] C. Albrecht, "Provably Good Routing by a New Approximation Algorithm for Multicommodity Flow", *Proc. ISPD*, 2000, pp. 19-25.
- [5] C. Albrecht, A. B. Kahng, I. Mandoiu and A. Zelikovsky, "Floorplan Evaluation with Timing-Driven Global Wireplanning, Pin Assignment, and Buffer/Wire Sizing", *Proc. ASP-DAC*, 2002, pp. 580-587.
- [6] C. J. Alpert, W.-K. Chow, K. Han, A. B. Kahng, Z. Li, D. Liu, S. Venkatesh, "Prim-Dijkstra Revisited: Achieving Superior Timing-driven Routing Trees", *Proc. ISPD*, 2018, to appear.
- [7] C. J. Alpert, R. G. Gandham, J. Hu, S. T. Quay and A. J. Sullivan, "Apparatus and Method for Determining Buffered Steiner Trees for Complex Circuits", *US Patent 6591411*, Jul. 2003.
- [8] C. J. Alpert, T. C. Hu, J. H. Huang, A. B. Kahng and D. Karger, "Prim-Dijkstra Tradeoffs for Improved Performance-driven Routing Tree Design", *IEEE TCAD* 14(7) (1995), pp. 890-896.
- [9] C. J. Alpert, J. Hu and P. H. Villarrubia, "Practical Methodology for Early Buffer and Wire Resource Allocation", *US Patent 6996512*, Feb. 2006.
- [10] C. J. Alpert and A. B. Kahng, "Multi-Way Partitioning Via Geometric Embeddings, Orderings, and Dynamic Programming", *IEEE TCAD* 14(11) (1995), pp. 1342-1358.
- [11] C. J. Alpert and A. B. Kahng, "A General Framework for Vertex Orderings, With Applications to Circuit Clustering", *IEEE TVLSI* 4(2) (1996), pp. 240-246.
- [12] C. J. Alpert, A. B. Kahng, C. N. Sze and Q. Wang, "Timing-driven Steiner Trees are (Practically) Free", *Proc. DAC*, 2006, pp. 389-392.
- [13] S. Bose, "Methods and Systems for Placement and Routing", *US Patent 8332793*, Dec. 2012.
- [14] U. Brenner, M. Struzyna and J. Vygen, "BonnPlace: Placement of Leading-edge Chips by Advanced Combinatorial Algorithms", *IEEE TCAD* 27(9) (2008), pp. 1607-1620.
- [15] U. Brenner and J. Vygen, "Worst-case Ratios of Networks in the Rectilinear Plane", *Networks* 38(3) (2001), pp. 126-139.
- [16] A. E. Caldwell, A. B. Kahng and I. L. Markov, "Can Recursive Bisection Alone Produce Routable Placements?", *Proc. DAC*, 2000, pp. 477-482.
- [17] A. E. Caldwell, A. B. Kahng and I. L. Markov, "Optimal Partitioners and End-case Placers for Standard-cell Layout", *IEEE TCAD* 19(11) (2000), pp. 1304-1313.
- [18] R. C. Carden, J. Li and C. K. Cheng, "A Global Router with a Theoretical Bound on the Optimal Solution", *IEEE TCAD* 15(2) (1996), pp. 208-216.
- [19] C.-K. Cheng, "Linear Placement Algorithms and Applications to VLSI Design", *Networks* 17(4) (1987), pp. 439-464.
- [20] C. K. Cheng, R. Graham, I. Kang, D. Park and X. Wang, "Tree Structures and Algorithms for Physical Design", *Proc. ISPD*, 2018.
- [21] C. K. Cheng, A. B. Kahng, I. Kang and L. Wang, "rePlace: Advancing Solution Quality and Routability Validation in Global Placement", *manuscript in submission*, 2017.
- [22] S. Chen and T. Yoshimura, "Fixed-Outline Floorplanning: Block-Position Enumeration and a New Method for Calculating Area Costs", *IEEE TCAD* 27(5) (2008), pp. 858-871.
- [23] J. Cong, A. B. Kahng, G. Robins and M. Sarrafzadeh, "Provably Good Performance-driven Global Routing", *IEEE TCAD* 11(6) (1992), pp. 739-752.
- [24] J. Cong and S. K. Lim, "Edge Separability-based Circuit Clustering with Application to Multi-level Circuit Partitioning", *IEEE TCAD* 23(3) (2004), pp. 346-357.
- [25] P. Dickinson, J. Hulshof and A. Ran, "Optimal Flood Control" (2011).
- [26] E. W. Dijkstra, "A Note on Two Problems in Connexion with Graphs", *Numerische Mathematik* 1 (1959), pp. 269-271.
- [27] Y. Du and M. D. F. Wong, "Optimization of Standard Cell Based Detailed Placement for 16nm FinFET Process", *Proc. DATE*, 2014, pp. 1-6.
- [28] M. Elkin and S. Solomon, "Narrow-Shallow-Low-Light Trees with and without Steiner Points", *SIAM J. Discrete Math.* 25(1) (2011), pp. 181-210.
- [29] G. M. Furnish, M. J. LeBrun and S. Bose, "Node Spreading Via Artificial Density Enhancement to Reduce Routing Congestion", *US Patent 7921392*, Apr. 2011.
- [30] G. M. Furnish, M. J. LeBrun and S. Bose, "Tunneling as a Boundary Congestion Relief Mechanism", *US Patent 7921393*, Apr. 2011.
- [31] R. E. Gomory and T. C. Hu, "Multi-Terminal Network Flows", *J. SIAM* 9(4) (1961), pp. 551-570.
- [32] C. Han, K. Han, A. B. Kahng, H. Lee, L. Wang and B. Xu, "Optimal Multi-Row Detailed Placement for Yield and Model-Hardware Correlation Improvements in Sub-10nm VLSI", *Proc. IC-CAD*, 2017, pp. 667-674.
- [33] L. He, S. Yao, W. Deng, J. Chen and L. Chao, "Interconnect Routing Methods of Integrated Circuit Designs", *US Patent 8386984*, Feb. 2013.
- [34] R. F. Heintschke, M. de Oliveira Johann, J. Narasimhan and R. A. de Luz Reis, "Methods and Apparatus for Providing Flexible Timing-driven Routing Trees", *US Patent 8095904*, Jan. 2012.
- [35] W. A. Horn, "Single-Machine Job Sequencing with Treelike Precedence Ordering and Linear Delay Penalties", *SIAM J. Appl. Math.* 32(2) (1972), pp. 189-202.
- [36] T. C. Hu, "The Maximum Capacity Route Problem", *Operations Research* 9(6) (1961), pp. 898-900.
- [37] T. C. Hu, "Multi-commodity Network Flows", *Operations Research* 11(3) (1963), pp. 344-360.
- [38] T. C. Hu, "On the Feasibility of Simultaneous Flows in a Network", *SIAM J.*, 12(2) (1964), pp. 359-360.
- [39] T. C. Hu, "Optimum Communication Spanning Trees", *SIAM J. Computing* 3(3) (1974), pp. 188-195.
- [40] T. C. Hu, *Combinatorial Algorithms*, Addison-Wesley, 1982.
- [41] T. C. Hu, "Math, Models and Methods", keynote address and paper, *Proc. 1st ACM Intl. Symp. on Physical Design*, April 1997, pp. 207-210.
- [42] T. C. Hu, A. B. Kahng and G. Robins, "Solution of the Discrete Plateau Problem", *Proc. National Academy of Sciences* 89(10), October 1992, pp. 9235-9236.
- [43] T. C. Hu, A. B. Kahng and G. Robins, "Optimal Robust Path Planning in General Environments", *IEEE Trans. Robotics and Automation* 9(6) (1993), pp. 775-784.
- [44] T. C. Hu and E. S. Kuh, eds., *VLSI Circuit Layout: Theory and Design*, New York, IEEE Press, 1985.
- [45] T. C. Hu and K. Moerder, "Multiterminal Flows in a Hypergraph", in: T. C. Hu and E. Kuh (Eds.), *VLSI Circuit Layout: Theory and Design* (IEEE Press, New York, 1985), pp. 87-93.
- [46] T. C. Hu and M. T. Shing, "The α - β Routing", in: T. C. Hu and E. Kuh (Eds.), *VLSI Circuit Layout: Theory and Design* (IEEE Press, New York, 1985), pp. 139-143.
- [47] T. C. Hu and M. T. Shing, *Combinatorial Algorithms, Enlarged Second Edition*, New York, Dover Publications, 2002.
- [48] E. Ihler, D. Wagner and F. Wagner, "Modeling Hypergraphs by Graphs with the Same Mincut Properties", *Information Processing Letters* 45(4) (1993), pp. 171-175.
- [49] A. B. Kahng, "classical Floorplanning Harmful?", *Proc. ISPD*, 2000, pp. 207-213.
- [50] A. B. Kahng, P. Tucker and A. Zelikovsky, "Optimization of Linear Placements for Wirelength Minimization with Free Sites", *Proc. ASP-DAC*, 1999, pp. 241-244.
- [51] J. M. Kleinbans, G. Sigl, F. M. Johannes and K. J. Antreich, "GORDIAN: VLSI Placement by Quadratic Programming and Slicing Optimization", *IEEE TCAD* 10(3) (1991), pp. 356-365.
- [52] G. Kortarz and D. Peleg, "Approximating Shallow-light Trees", *Proc. SODA*, 1997, pp. 103-110.
- [53] T. Leighton and S. Rao, "An Approximate Max-Flow Min-Cut Theorem for Uniform Multicommodity Flow Problems with Applications to Approximation Algorithms", *Foundations of Computer Science*, 1988, pp. 422-431.
- [54] C.-T. Lin, D.-S. Chen and Y.-W. Wang, "Fixed-Outline Floorplanning Through Evolutionary Search", *Proc. ASP-DAC*, 2004, pp. 42-44.
- [55] Y. Lin, B. Yu, X. Xu, J.-R. Gao, N. Viswanathan, W.-H. Liu, Z. Li, C. J. Alpert and D. Z. Pan "MRDP: Multiple-row Detailed Placement of Heterogeneous-sized Cells for Advanced Nodes", *Proc. ICCAD*, 2016, pp. 7-1-7-8.
- [56] Y. Lin, B. Yu, B. Xu and D. Z. Pan, "Triple Patterning Aware Detailed Placement Toward Zero Cross-Row Middle-of-Line Conflict", *Proc. ICCAD*, 2015, pp. 396-403.
- [57] R. Liu, S. Dong, X. Hong and Y. Kajitani, "Fixed-Outline Floorplanning with Constraints Through Instance Augmentation", *Proc. ISCAS*, 2005, pp. 1883-1886.
- [58] C. Luo, M. F. Anjos and A. Vannelli, "Large-Scale Fixed-Outline Floorplanning Design Using Convex Optimization Techniques", *Proc. ASP-DAC*, 2008, pp. 198-203.
- [59] G. K. Moy, "A Specific Network Link and Path Likelihood Prediction Tool", *AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH*, 1996, No. AFIT/GCS/ENG/96D-21.
- [60] H. Nagamochi and T. Ibaraki, "Computing Edge-connectivity in Multigraphs and Capacitated Graphs", *SIAM J. Discrete Math.* 5(1) (1992), pp. 54-66.
- [61] R. C. Prim, "Shortest Connecting Networks and Some Generalizations", *Bell System Tech. J.* 36 (1957), pp. 1389-1401.
- [62] P. Saxena, V. Khandelwal, C. Qiao, P.-H. Ho, J. C. Lin and M. A. Iyer, "Interconnect-driven Physical Synthesis using Persistent Virtual Routing", *US Patent 7853915*, Dec. 2010.
- [63] M. Stoer and F. Wagner, "A Simple Min Cut Algorithm", in *European Symposium on Algorithms*, Berlin, Springer, pp. 141-147.
- [64] A. Surampudi and K. Kalimuthu, "An Energy Efficient Spectrum Sensing in Cognitive Radio Wireless Sensor Networks" *arXiv preprint arXiv:1711.09255* (2017).
- [65] R. S. Tsay, E. S. Kuh and C. P. Hsu, "PROUD: A Sea-of-Gates Placement Algorithm", *IEEE Design and Test of Computers* 5(6) (1988), pp. 44-56.
- [66] J. Z. Yan and C. Chu, "DeFer: Deferred Decision Making Enabled Fixed-Outline Floorplanning Algorithm", *IEEE TCAD* 29(3) (2010), pp. 367-381.
- [67] H. H. Yang and D. F. Wong, "Efficient Network Flow Based Min-Cut Balanced Partitioning", *IEEE TCAD* 15(12) (1996), pp. 1533-1540.
- [68] H. Zou, W. Zhang and Q. Wang, "An Improved Cerebral Vessel Extraction Method for MRA Images", *Bio-medical Materials and Engineering* 26(s1) (2015), pp. S1231-S1240.