On Algorithmic Applications of the Immersion Order*

An Overview of Ongoing Work Presented at the Third Slovenian International Conference on Graph Theory

Michael A. Langston and Barbara C. Plaut
Department of Computer Science
University of Tennessee
Knoxville, TN 37996–1301

Abstract

A snapshot of our current exploration of the algorithmic aspects of the immersion order is presented. Integrated circuit partitioning is used as a prototypical applications domain. Decision and search algorithms, self-reductions, closure-preserving operators and related developments are discussed.

* This research is supported in part by the National Science Foundation under grant CDA–9115428 and by the Office of Naval Research under contracts N00014–90–J–1855 and N00014–94–1–1155.

1 Background

We consider only finite, undirected graphs. H is said to be *immersed* in G, written $H \leq_i G$, iff a graph isomorphic to H can be obtained from G by lifting pairs of adjacent edges and taking a subgraph. As an example, observe that C_4 is immersed in $K_1 + 2K_2$.

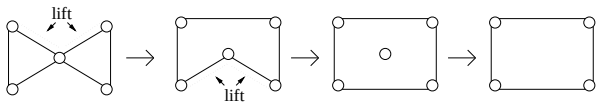

Figure 1: $C_4 \leq_i K_1 + 2K_2$

Suppose a family F is *closed* in this order, that is, $G \in F$ and $H \leq_i G \Rightarrow H \in F$. The *obstruction set* for F consists of the immersion-minimal elements in F's complement. Accordingly, F has the following characterization: G is in F iff no obstruction for F is immersed in G. It is known [RS1] that any such obstruction set is finite. It is also known [FL3, RS2] that deciding whether $H \leq_i G$ is decidable in polynomial time for every fixed H. Thus, there exists a polynomial-time recognition algorithm for any immersion-closed family of graphs. Such an algorithm is not constructively known, but possesses a time bound of $O(n^{h+3})$, where h denotes the order of the largest obstruction.

One of the earliest and best-known applications of the immersion order is the *min cut linear arrangement* problem. Though $\mathcal{NP}$-complete in general, the fixed-parameter version of this problem has been shown to be decidable in linear time with the aid of the immersion order and special tools based on the treewidth metric [FL3, Bo]. Much less is known, however, about the vast majority of applications.

2 Circuit Partitioning

Consider the field programmable gate array (henceforth FPGA), a collection of logic blocks with programmable connections [MNSBS]. A given circuit is implemented by partitioning

its logic into blocks and connecting the blocks as required.

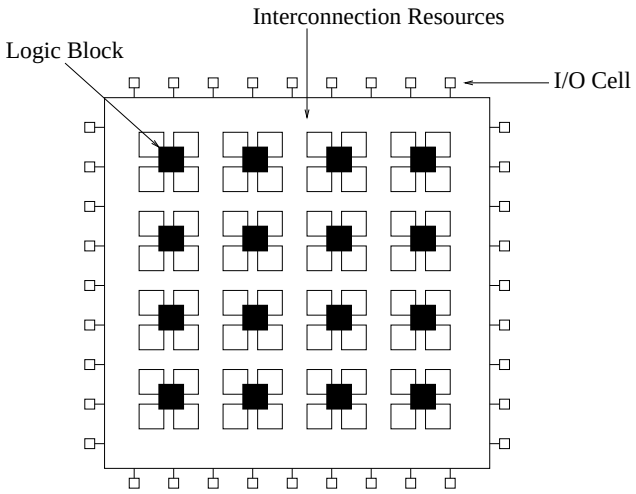

Figure 2: The FPGA

Since circuits are frequently too large to fit on a single chip, they must be partitioned over several FPGA's. In building systems with multiple FPGA's, fabrication technology imposes severe restrictions: limits on pin counts (I/O cells) affect inter-chip connectivity; limits on chip area and density bound FPGA sizes.

Such practical limitations motivate many interesting combinatorial problems. Consider, for example, the Min Degree Graph Partition problem. In this problem, we are given a graph $G = (V, E)$ and two integers k and d, and are asked whether V can be partitioned into disjoint subsets $V_1, V_2, ..., V_m$ so that, for $1 \leq i \leq m$, $|V_i| \leq k$ and at most d edges have exactly one end-point in V_i. In a multi-FPGA context, for example, G models the circuit to be partitioned, k denotes the maximum number of logic blocks permitted on a chip, and d represents the maximum degree or pin count of any chip.

This problem is clearly very difficult, in fact intractable without parameter bounds, via a reduction from Multiway Cut [Go].

Theorem *Min Degree Graph Partition is $\mathcal{NP}$-complete.*

Fortunately, however, the aforementioned fabrication limits can be used to advantage. As long as k and d are bounded, the family of "yes" instances is closed in the immersion order.

Theorem *For any fixed k and d, Min Degree Graph Partition can be decided in polynomial time.*

The last statement is of particular interest in light of the observation that, unlike Multiway Cut, Min Degree Graph Partition has no known brute-force polynomial-time algorithm when k and d are fixed. This is in contrast to the superficially-similar Graph Partition problem, in which the cost of a solution is summed over all subsets rather than measured over each, thus bounding the maximum number of partitions.

Results such as this inherently rely on the existence of finite lists of immersion-minimal obstructions. As of this writing, little is known about such obstructions in general or about practical immersion tests in particular. As with the minor order, we expect that even partial sets can be useful [Ra]. It has been observed that complete graphs are often obstructions to immersion-closed families. Testing for K_1, K_2 or K_3 is easy. Testing for K_4 turns out to be quite complicated, however, though achievable in linear time. See [BGLR] for decision, search and parallel algorithms.

Min Degree Graph Partition is an excellent example of the current state of the art. We have identified a wide array of other problems, largely from the circuit partitioning domain, amenable to tools based on the immersion order. For most of these, just as with Min Degree Graph Partition, we can at present say not much more than that they are (nonconstructively) decidable in polynomial time. Whether they are solvable in low-order polynomial time, perhaps even linear time, is an open question, and one we are actively pursuing. One might be tempted to employ the treewidth metric, useful for Min Cut Linear Arrangement. If the family of "yes" instances has bounded treewidth, linear time recognizability is assured. But that is not generally the case. To see this, consider Min Degree Graph Partition with $k = 1$ and $d = 4$. Even this simple family of graphs contains the $w \times w$ grid for any w, a graph with

treewidth w. One might also ask about eliminating nonconstructivity. We have developed some techniques for that task, although they are mainly of theoretical interest and beyond the scope of this brief review. We refer the reader to [FL4] for details.

3 Search Algorithms and Self-Reducibility

It is sometimes possible to solve a search problem by reducing it to a related decision problem. For example, one might seek to find a satisfying subset assignment for Min Degree Graph Partition with the aid of a routine that merely tells whether such an assignment exists.

This approach to algorithm design is called *self-reducibility*, and has been formulated in many ways in the literature. In its most limited form, an assortment of restrictions are placed on the decision algorithm, its input and the lexicographic position of the output produced (see, for example, [Sc]). In more general forms, input/output limitations are eliminated and decision algorithms quite distant from the original problem are permitted (see, for example, [FL2]). Additional variations exist, some even incorporating randomness or parallelism (see, for example, [FF, KUW]).

It is not difficult to see that, for any fixed k and d, Min Degree Graph Partition is self-reducible in polynomial time. That is, one can construct a satisfying subset assignment, if any exist, with at most a polynomial number of calls to a decision algorithm, known from the last section also to run in polynomial time.

It can in fact be self-reduced with only a linear number of calls. No vertex in a "yes" instance has $d+k$ or more neighbors (a star with $d+k$ rays is an obstruction). Furthermore, in such an instance, there must exist some satisfying assignment in which each subset induces a connected subgraph. From this it can be shown that, no matter the rest of the partition, two vertices not connected by a sufficiently short path need never share the same subset. Thus we know in advance that, as a solution is recursively constructed, a vertex v need share a subset only with candidates from a bounded-size neighborhood. Each such candidate, u, can be tested for suitability by adding $d+1$ copies of the edge uv, calling the decision algorithm

and retaining the extra edges only when the resulting graph is also a "yes" instance.

Theorem *For any fixed k and d, the search version of Min Degree Graph Partition can be solved in $O(np(n))$ time, where $p(n)$ denotes the time required to solve the decision version of the problem.*

A number of interesting self-reducibility issues remain open for this order, though none yet are perhaps as noteworthy as embedding reducibilities are for the minor order [FL4].

4 Closure-Preserving Operators

In the case of a "no" instance, some sort of approximation scheme is often required. But increasing the size of problem parameters may not be desirable or even possible in many settings. An approach with some practical appeal then is to ask instead whether one can modify the graph (simplify the underlying circuit) so that it becomes a "yes" instance. More generally, we seek systematic methods for making such modifications so as to preserve immersion closure.

Let F denote a family of graphs, and let $F_v(h)$ denote those graphs for which there exists some set of h or fewer vertices whose removal creates a graph in F. When h is fixed, recognizing $F_v(h)$ can of course be reduced to recognizing F by brute force in time proportional to n^h, a polynomial. If F is minor-closed, however, there is a more efficient technique. It is known [FL1] that if F is minor-closed, then so is $F_v(h)$.

Unfortunately, this operator does not work for the immersion order. To see this, let F denote the family of edgeless graphs, and let $h = 1$. The star graph with three rays is in $F_v(1)$, but the graph obtained by lifting a pair of edges yields a matching of size two, which is not in $F_v(1)$.

So consider edges instead, and let $F_e(h)$ denote those graphs for which there exists some set of h or fewer edges whose removal creates a graph in F.

Theorem *For any fixed h, if F is immersion-closed, then so is $F_e(h)$.*

This operator, plus self-reducibility, therefore yields a polynomial-time approach for solving the decision and search versions of $F_e(h)$ when, for example, F denotes Min Degree Graph Partition. Other operators exist, but this is perhaps the most natural from an algorithmic standpoint.

5 In Closing

Much is known about complexity-theoretic issues for subgraph, topological and even minor containment [BL]. In contrast, we have thus far really only scratched the surface in understanding some of the range and depth of algorithmic applications of the immersion order. Many challenging open questions beckon, several of which we have attempted to illuminate here.

References

[BGLR] H. Booth, R. Govindan, M. A. Langston, and S. Ramachandramurthi, "Sequential and Parallel Algorithms for K_4 Immersion Testing," Technical Report, University of Tennessee, 1995.

[BL] D. Bienstock and M. A. Langston, "Algorithmic Implications of the Graph Minor Theorem," in Handbook of Operations Research and Management Science: Network Models (M. O. Ball, T. L. Magnanti, C. L. Monma and G. L. Nemhauser, editors), North–Holland, 1995, 481–502.

[Bo] H. L. Bodlaender, "A Linear Time Algorithm for Finding Tree-Decompositions of Small Treewidth," Technical Report, Utrecht University, 1992.

[FF] J. Feigenbaum and L. Fortnow, "Random Self-reducibility of Complete Sets," *SIAM Journal on Computing* 22 (1993), 994–1005.

[FL1] M. R. Fellows and M. A. Langston, "Nonconstructive Tools for Proving Polynomial-Time Decidability," *Journal of the ACM* 35 (1988), 727–739.

[FL2] M. R. Fellows and M. A. Langston, "Fast Search Algorithms for Layout Permutation Problems," *International Journal of Computer Aided VLSI Design* 3 (1991), 325–342.

[FL3] M. R. Fellows and M. A. Langston, "On Well-Partial-Order Theory and Its Application to Combinatorial Problems of VLSI Design," *SIAM Journal on Discrete Mathematics* 5 (1992), 117–126.

[FL4] M. R. Fellows and M. A. Langston, "On Search, Decision and the Efficiency of Polynomial-Time Algorithms," *Journal of Computer and Systems Sciences* 49 (1994), 769–779.

[Go] R. Govindan, "On the Complexity of Circuit Partitioning," Ph.D. Dissertation, University of Tennessee, 1995.

[KUW] R. M. Karp, E. Upfal and A. Wigderson, "The Complexity of Parallel Search," *Journal of Computer and Systems Sciences* 36 (1988), 225–253.

[MNSBS] R. Murgai, Y. Nishizaki, N. Shenoy, R. K. Brayton and A. Sangiovanni-Vincentelli, "Logic Synthesis for Programmable Gate Arrays," *Proceedings, 27th ACM/IEEE Design Automation Conference* (1990), 620–625.

[Ra] S. Ramachandramurthi, "Algorithms for VLSI Layout Based on Graph Width Metrics," Ph.D. Dissertation, University of Tennessee, 1994.

[RS1] N. Robertson and P. D. Seymour, "Graph Minors IV. Tree-width and Well-quasi-ordering," *J. Combin. Theory Ser. B* 48 (1990), 227–254.

[RS2] N. Robertson and P. D. Seymour, "Graph Minors XIII. The Disjoint Paths Problem," *J. Combin. Theory Ser. B* 63 (1995), 65–110.

[Sc] C. P. Schnorr, "Optimal Algorithms for Self-reducible Problems," *Proceedings, 1976 International Conference on Automata, Programming and Languages* (1976), 322–337.