

# The Parallel Complexity of Minimizing Column Conflicts\*

John E. Savage  
Brown University  
Department of Computer Science  
Providence, Rhode Island 02912  
Tel: (401) 863-7600, Fax: (401) 863-7657  
Email: jes@cs.brown.edu

Markus G. Wloka  
Motorola, Inc., MD EL 510,  
SSDT System and Behavioral Design Group  
2100 E. Elliot Road, Tempe, AZ 85284  
Tel: (602) 897-4452, Fax: (602) 897-3518  
Email: mgw@ssdt-tempe.sps.mot.com

## Abstract

*Two-layer channel routers typically require a post-processing phase to reduce or eliminate column conflicts. Attempts have been made to parallelize this problem using local search heuristics that swap horizontal channel wire segments. We show that all such heuristics for this problem are P-hard and unlikely to be efficiently parallelizable.*

## 1.0 Introduction

We examine local search algorithms for the minimization of column conflicts, a key phase of the channel-routing problem. We show that any such algorithms based on cost-improving swaps of horizontal tracks in a channel are P-hard. Thus, they are at least as hard to parallelize as any polynomial time problem.

Although, as we have shown in [SW88], the “left-edge” channel-routing algorithm can be parallelized to run in polylogarithmic time with a polynomial number of processors, it does not deal with column conflicts. Therefore, it is tempting to design a parallel heuristic based on local search that minimizes column conflicts or tries to remove them altogether. We are motivated by the work of Brouwer and Banerjee [BJS90] who show that parallel simulated annealing on a few processors minimizes column conflicts and yields routings of high quality. We are interested in very large problems executed on very highly parallel machines, cases not considered by them.

---

\* This work was supported in part by the National Science Foundation under Grant CDA 87-22809 and the Office of Naval Research under contract N00014-83-K-0146 and ARPA Order Nos. 4786, 6320.

In two-layer channel routing a net is implemented by one horizontal wire in one layer and vertical wires linking the horizontal one to the terminals in the other layer. *Vias* connect wires in different layers and a *column conflict* occurs if two terminals on opposite sides of a channel have overlapping vertical wires. The goal is to produce a routing without conflicts while minimizing the number of tracks.

In the work mentioned above, [SW88], we develop an optimal  $O(\log n)$  time,  $O(n)$  processor EREW PRAM parallel version of the “left-edge” algorithm developed by Hashimoto and Stevens [HS71] for PCB routing. Our algorithm achieves channel density, when there are no column conflicts, and was implemented on the CM-2 Connection Machine where it runs in time  $O(\log^2 n)$  with  $O(n)$  processors.

Many variations and extensions of the basic left-edge algorithm have been proposed to deal with column conflicts, all of which are serial, namely, Deutsch’s dogleg channel router [Deutsch76], Dolev *et al.*’s channel router [DKSSU81], Fiduccia and Rivest’s greedy channel router [RF82], Yoshimura and Kuh’s channel router [KY82], and YACR2 by Reed *et al.* [RSS85].

These heuristics for two-layer channel routing with column conflicts produce routings of very good quality but are not guaranteed to achieve minimum channel width or channel density. Most of the more general two-layer channel routing problems are NP-complete, as shown by LaPaugh [Lapaugh80], Szymanski [Szymanski85], and Sarrafzadeh [Sarrafzadeh87].

## 2.0 Local Search Heuristics for Column Conflict Reduction

*Local search heuristics* have been established as the method of choice in many optimization problems whenever very good solutions are desired. In local search heuristics, an initial solution  $S_1$  is constructed, usually by

some random procedure, and a cost function  $f$  is computed. Changes are made to the current solution and a new solution  $S_2$ , which we say is in the *neighborhood* of the current solution, replaces the current solution. The improvement in the cost function  $f$  obtained from changing from solution  $S_1$  to solution  $S_2$  is given by the *gain function*  $\Delta f = f(S_1) - f(S_2)$ . A positive gain change is one that decreases cost. The termination rules differ for each heuristic.

Classical local search heuristics are *steepest descent*, the heuristics of *Kernighan-Lin* (KL) [KL70], *Fiduccia-Mattheyses* (FM) [FM82] and *simulated annealing* (SA) [KGV83], inspired by the work of Metropolis *et al.* [MRRTT53] on statistical mechanics.

SA selects at random a solution in the neighborhood of the current solution, computes the gain  $\Delta f$  and accepts that solution if it reduces the cost. If not, the solution is accepted with a probability that decreases (usually exponentially) with the size of the increase in cost. This probability function also decreases with the “temperature” of the annealing process which decreases with time. The high-quality results given by SA are explained by observing that it appears to find the region of a global minimum quickly and then homes in on the minimum as the temperature decreases.

Since local search problems require considerable computational resources, recent work has addressed parallel methods. Most of this work has focused on parallel simulated annealing. (See Greening [Greening90] for a survey of parallel simulated-annealing techniques.) The simplicity of the SA algorithm seems to make an implementation on a massively parallel computer almost trivial. It is therefore surprising that the goal of parallelizing SA by straightforward or elaborate means has eluded researchers. We have shown that for a number of other problems related to VLSI placement, including graph-partitioning [SW90], [SW91b] and grid and hypercube embedding [SW91a], [SW91c], local search heuristics based on the SWAP-neighborhood are P-complete or P-hard. This class of heuristics includes SA at zero temperature. Schaeffer and Yannakakis [SY91] have independently shown that local search under the SWAP neighborhood for graph partitioning is P-complete. We show in this paper that the P-hardness results for local search also apply to column conflict minimization.

To complete the description as a local search problem of the minimization of column conflicts in channel routing, we need a *cost function* and a method for moving from one solution to another within a *neighborhood*.

**Definition 1** The column cost  $\text{COLUMNS}(R)$  of a two-layer channel routing  $R$  (a mapping of nets to tracks) is the number of overlapping vertical segments.

The goal, which may not be achievable, is to reduce the COLUMNS cost of zero. Such solution may be hard to find with a local search heuristic or may not exist at all. A solution with low COLUMNS cost is desirable even if it is not routable in two layers with one horizontal wire segment per net. The YACR2 channel router [RSS85] and the router by Brouwer and Banerjee [BJS88] use a post-processing pass with dog-legging and the insertion of extra tracks to remove remaining conflicts from a channel routing with low COLUMNS cost.

Given a two-layer channel routing, a natural way to generate another routing with potentially fewer of column conflicts is to swap two horizontal wire segments in different tracks which when swapped do not overlap with other horizontal wires. Such swaps, which we call *subtrack swaps*, do not affect the number of tracks but may reduce the number of nets whose overlapping vertical wire segments.

**Definition 2** The SUBTRACK-SWAP( $R$ ) neighborhood of a channel-routing solution  $R$  is the set of channel routings obtained by swapping a pair of horizontal wires.

### 3.0 Parallel Complexity

**Definition 3** Let  $P$  be the class of polynomial-time decision problems. A decision problem  $A \in P$  is *logspace-reducible* to a problem  $B$  if there is a function  $g; \{0, 1\}^* \rightarrow \{0, 1\}^*$  computable in logarithmic space (logspace) by a deterministic Turing machine such that  $x \in A$  if and only if  $g(x) \in B$ . A decision problem is *P-hard* if every problem in  $P$  is logspace-reducible to it.

**Definition 4** The circuit-value problem (CVP) is the problem of computing the value of a Boolean circuit [Savage76] from a description of the circuit and values for its inputs.

**Theorem 1** CVP is P-complete [Ladner75].

Restricted versions of CVP are also P-complete. A *monotone circuit* uses only the operations AND and OR.

**Corollary 1** The monotone circuit-value problem (MCVP) is P-complete [Goldschlager77].

The *ordered monotone circuit-value problem* (OMCVP) is the problem of computing the value of a monotone

Boolean circuit  $OM(C)$  from a description of the circuit and values for its inputs. The circuit elements in  $OM(C)$  are indexed so that when circuit element  $k$  depends on gate  $i$ , then  $k > i$ . The following theorem is a trivial extension of the theorems of Ladner [Ladner75] and Goldschlager [Goldschlager77].

**Corollary 2** *The ordered circuit-value problem OMCVP is P-complete.*

#### 4.0 P-hardness of the Column-Swap Heuristic

In this section we sketch our proof of the following theorem which asserts that finding a locally minimum channel routing using the COLUMNS cost function and the SUBTRACK-SWAP neighborhood is P-hard. The theorem applies to deterministic and randomized algorithms. For the latter, acceptance is probabilistic.

**Theorem 3** *Let H be a local search algorithm for channel routing using the COLUMNS cost function and the SUBTRACK-SWAP neighborhood. If H accepts only swaps that improve the cost, then H is P-hard.*

Our proof consists of a logspace procedure to translate a given fan-in/fan-out 2, ordered monotone circuit  $OM(C)$  with  $n$  gates into an initial solution  $R_0$  of a channel-routing problem. When we apply heuristic H to  $R_0$ , it executes a polynomial number  $q(n)$  of swaps, returns a solution  $R_{q(n)}$  with a locally minimum number of column conflicts from which the value of  $OM(C)$  can be determined in logarithmic space. Thus H is P-hard.

**Construction of Circuit  $OM(C)$**  We simulate a circuit with nets that computes exactly the values of an arbitrary circuit  $OM(C)$ . Sets of nets simulate inputs, gates and the wires between gates in the circuit  $OM(C)$ .

Figure 1 shows the overall construction of a channel routing problem from the circuit  $OM(C)$ . Figures 2, 3, 4, and 5 show the components of the channel routing problem. The numbers above vertical columns are their width, which is actually the number of individual vertical segments of which they are made. If two wires of width  $k$  overlap, the column cost is  $k$ .

**FIGURE 1. The overall construction of a channel routing problem from the circuit  $OM(C)$**

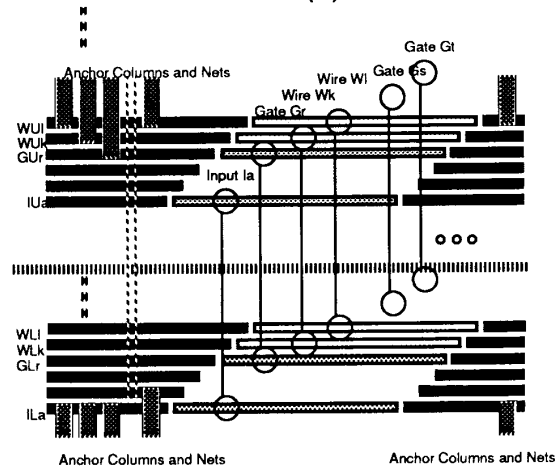


Figure 2 shows how a wire that connects gates is built out of nets. The wire  $W_k$  consists of one pair  $WL_k, WU_k$ . Figure 2 also shows the pair  $GL_r, GU_r$  of the gate  $G_r$  and the pair  $GL_s, GU_s$  of the gate  $G_s$ .

**FIGURE 2. Connecting wire  $W_k$**

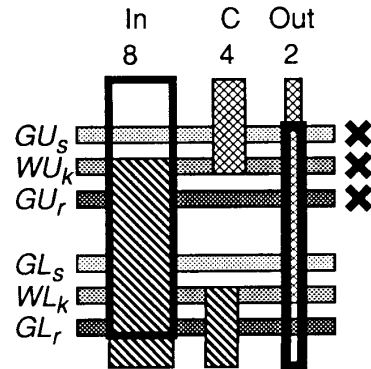


FIGURE 3. Circuit input  $I_a$ . (a)  $I_a = 0$ . (b)  $I_a = 1$

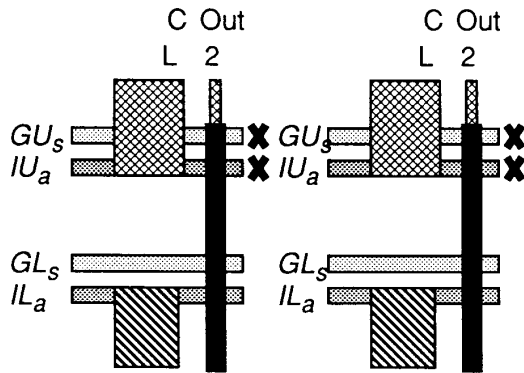


Figure 3 shows how circuit inputs are built out of nets. Circuit input  $I_a$  consists of one pair, the center pair  $IL_a$ ,  $IU_a$ . To demonstrate how circuit inputs are attached to gates, Figure 3 also shows the pairs  $GL_s$ ,  $GU_s$  of the gate  $G_s$ . The problem description of the circuit  $OM(C)$  gives the boolean values of the circuit inputs and specifies that every circuit input is used exactly once. When input  $I_a = 0$ , one column of weight 2 is attached from the top of the channel to  $IL_a$ . When  $I_a = 1$ , one column of weight 2 is attached from the top of the channel to  $IU_a$ . (The black-bordered column of weight 2 is part of gate  $G_s$ , and is included in Figure 3 to show how the circuit is assembled.) The columns attached to the center pair  $IL_a$ ,  $IU_a$  have a weight that is made so large that swapping of pair  $IL_a$ ,  $IU_a$  causes a negative gain that is greater than the sum of the weights on all AND/OR gates and connecting wires in the routing of  $OM(C)$ .

FIGURE 4. OR gate  $G_r$   
In1 In2 C Out1 Out2  
2 2 1 8 8

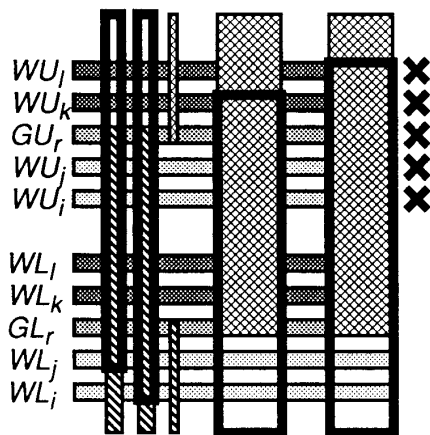
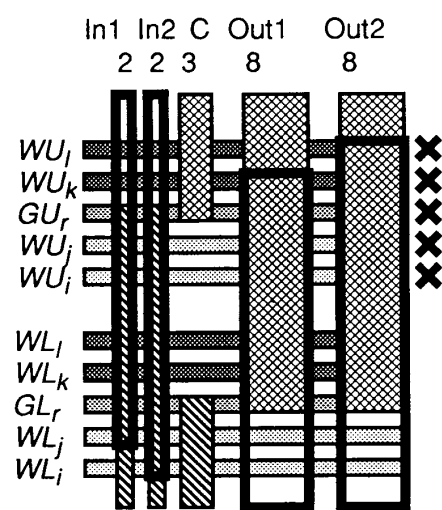


FIGURE 5. AND gate  $G_r$   
In1 In2 C Out1 Out2  
2 2 3 8 8



Figures 4 and 5 show the construction of AND and OR gates from nets. Both AND and OR gates  $G_r$  contain the center subtrack pair  $GL_r$ ,  $GU_r$ , which determines the state of the gate. Figures 4 and 5 also show the two pairs  $WL_i$ ,  $WU_i$  and  $WL_j$ ,  $WU_j$  of the wires  $W_i$  and  $W_j$ , which serve as inputs to gate  $G_r$ , and the pairs  $WL_k$ ,  $WU_k$  and  $WL_l$ ,  $WU_l$  of the wires  $W_k$  and  $W_l$ , which are the two copies of the output of gate  $G_r$ .

**Operation of Circuit  $OM(C)$**  Pairs of nets associated with values of inputs, gates or wires will either remain in their initial position, and said to have *value 0*, or they will swap, and said to have *value 1*. The routing problem that we construct is designed to insure that input values propagate, just as they do in gates, from inputs to outputs via wires. A detailed analysis is needed of all circuit components to show that they interact properly and compute the value of the circuit  $OM(C)$  which they simulate. The complete proof is given in [Wloka91].

## 5.0 Conclusions

We have outlined our proof that any local search heuristic based on cost-improving swaps of tracks or subtracks that minimizes the number of column conflicts in a two-layer channel routing by accepting is P-hard. It would be interesting to know whether Fiduccia and Rivest's greedy channel router [RF82] and YACR2 by Reed *et al.* [RSS85] are also hard to parallelize.

## Bibliography

- [BJS90] P. Banerjee, M. Howard Jones, J. S. Sargent, "Parallel Simulated Annealing Algorithms for Cell Placement on Hypercube Multiprocessors," *IEEE Transactions on Parallel and Distributed Systems*, vol. PDS-1, no. 1, pp. 91-106, January 1990.
- [Deutsch76] D. N. Deutsch, "A Dogleg Channel Router," *Proceedings 13th IEEE Design Automation Conference*, pp. 425-433, 1976.
- [DKSSU81] D. Dolev, K. Karplus, A. Siegel, A. Strong, J.D. Ullman, "Optimal Wiring between Rectangles," *13th Annual ACM Symposium on the Theory of Computing*, pp. 312-317, 1981.
- [FM82] C. M. Fiduccia, R. M. Mattheyses, "A Linear-Time Heuristic for Improving Network Partitions," *19th IEEE Design Automation Conference*, pp. 175-181, 1982.
- [Goldschlager77] L. M. Goldschlager, "The Monotone and Planar Circuit Value Problems," *ACM Sigact News*, vol. 9, no. 2, pp. 25-29, 1977.
- [Greening90] D. R. Greening, "Parallel Simulated Annealing Techniques," *Physica*, D 42, pp. 293-306, 1990.
- [HS71] A. Hashimoto, J. Stevens, "Wire Routing by Optimizing Channel Assignments Within Large Apertures," *Proceedings 6th IEEE Design Automation Conference*, pp. 155-163, 1971.
- [KL70] B. W. Kernighan, S. Lin, "An Efficient Heuristic Procedure for Partitioning Graphs," *AT&T Bell Labs Technical Journal*, vol. 49, pp. 291-307, February 1970.
- [KGV83] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, no. 4598, pp. 671-680, May 1983.
- [KY82] E. S. Kuh, T. Yoshimura, "Efficient Algorithms for Channel Routing," *IEEE Transactions on Computer-Aided Design*, pp. 25-35, vol. CAD-1, no. 1, January 1982.
- [LaPaugh80] A. S. LaPaugh, "Algorithms for Integrated Circuit Layout: an Analytic Approach," Ph.D. Thesis, MIT/LCS/TR-248, Massachusetts Institute of Technology, Cambridge, Ma., 1980.
- [Ladner75] R. E. Ladner, "The Circuit Value Problem is Log Space Complete for P," *ACM SIGACT News*, vol. 7, no. 1, pp. 18-20, 1975.
- [MRRTT53] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, E. Teller, "Equations of State Calculations by Fast Computing Machines," *Journal of Chemical Physics*, no. 21, vol. 6, pp. 1087-1091, June 1953.
- [RSS85] J. Reed, A. Sangiovanni-Vincentelli, M. Santomauro, "A New Symbolic Channel Router: YACR2," *IEEE Transactions on Computer-Aided Design*, vol. CAD-4, no. 3, pp. 208-219, July 1985.
- [RF82] R. L. Rivest, C. M. Fiduccia, "A Greedy Channel Router," *Proceedings 19th IEEE Design Automation Conference*, pp. 418-424, 1982.
- [Sarrafzadeh87] M. Sarrafzadeh, "Channel-Routing Problem in the Knock-Knee Mode is NP-Complete," *IEEE Transactions on Computer-Aided Design*, pp. 503-506, vol. CAD-6, no. 4, July 1987.
- [Savage76] J. E. Savage, *The Complexity of Computing*, John Wiley and Sons, 1976.
- [SW88] J. E. Savage, M. G. Wloka, "A Parallel Algorithm for Channel Routing," *Graph-Theoretic Concepts in Computer Science*, Lecture Notes in Computer Science, Springer-Verlag, Amsterdam, no. 344, pp. 288-301, June 1988.
- [SW90] J. E. Savage, M. G. Wloka, "On Parallelizing Graph-Partitioning Heuristics," *Proceedings of the ICALP'90*, Lecture Notes in Computer Science, Springer-Verlag, no. 443, pp. 476-489, July 1990.
- [SW91a] J. E. Savage, M. G. Wloka, "Parallelizing SA For Graph Embedding Is Hard," *Proceedings of the IMACS World Congress on Computation and Applied Mathematics*, Special Session on Simulated Annealing, Dublin, July 1991.
- [SW91b] J. E. Savage, M. G. Wloka, "Parallelism in Graph-Partitioning," *Journal of Parallel and Distributed Computing*, vol. 13, pp. 257-272, November 1991.
- [SW91c] J. E. Savage, M. G. Wloka, "Parallel Graph-Embedding and the Mob Heuristic," submitted, 1991.
- [SY91] A. A. Schaeffer, M. Yannakakis, "Simple Local Search Problems That Are Hard to Solve," *SIAM Journal on Computing*, vol. 20, no. 1, pp. 56-87, February 1991.
- [Szymanski85] T. G. Szymanski, "Dogleg Channel Routing is NP-Complete," *IEEE Transactions on Computer-Aided Design*, vol. CAD-4, no. 1, pp. 31-40, January 1985.
- [Wloka91] M. G. Wloka, "Parallel VLSI Synthesis," Ph.D. Thesis, Technical Report No. CS-91-35, Department of Computer Science, Brown University, RI, 1991.