

Techniques for Crosstalk Avoidance in the Physical Design of High-Performance Digital Systems*

Desmond A. Kirkpatrick

Alberto L. Sangiovanni-Vincentelli

Department of EECS, University of California, Berkeley, CA 94720

Abstract

Interconnect performance does not scale well into deep submicron dimensions, and the rising number of analog effects erodes the digital abstraction necessary for high levels of integration. In particular, crosstalk is an analog phenomenon of increasing relevance. To cope with the increasingly analog nature of high-performance digital system design, we propose using a constraint-driven methodology. In this paper we describe new constraint generation ideas incorporating digital sensitivity. In constraint-driven synthesis, we show that a fundamental subproblem of crosstalk channel routing, coupling-constrained graph levelization (CCL), is NP-complete, and develop a novel heuristic algorithm. To demonstrate the viability of our methodology, we introduce a gridless crosstalk-avoiding channel router as an example of a robust and truly constraint-driven synthesis tool.

1 Introduction

Increasingly, analog effects limit system performance as process dimensions scale into deep submicron, clock frequencies move into the hundreds of megahertz, and chip dimensions and integration levels continue to increase. While we still depend on process scaling to improve performance, if we ignore these second order effects we lose much of its benefit. Crosstalk noise is a particular concern as it can lead to delay and logic hazards, especially in dynamic circuitry useful to high performance circuit design.

Synthesis for performance has become a central theme in design automation for digital circuits, and a large amount of CAD research addresses first order interconnect performance issues. Yet the ability to control analog parasitic effects will likely determine the ultimate performance achievable in complex high-performance systems. In addition, though analog parasitic effects which break down this abstraction are becoming well understood, this understanding has not yet evolved to the point of controlled synthesis. Because these effects are second order, unlike parasitics such as capacitance or even RC delay, their impact on performance can only be controlled: optimization provides no real benefit.

Recent efforts in synthesis for crosstalk include post-route optimization [1] and compaction [2], both of which are not constraint-driven, but rather aim to reduce crosstalk. Conversely, there has been much work recently in analog CAD where control of second order effects is paramount to successful system design. A complete methodology for top-down constraint-driven synthesis is presented in [3]. An analog channel router in [4] handles pairwise coupling constraints derived from analog sensitivity analysis. Our work extends this work by introducing digital sensitivity and handling more general bounds, adding flexibility for synthesis.

In this paper we first discuss constraint generation, introducing a new digital sensitivity concept to complement analog sensitivity ideas. Here, we reduce the number of constraints so as to cope with the complexity of digital systems. Next we formulate the problem

of crosstalk-constrained channel routing to maximize router flexibility. We show that a subproblem of this formulation, levelizing a graph with coupling constraints, is NP-complete. Therefore, we turn to a novel heuristic method for partitioning these coupling constraints while routing, resulting in an efficient channel routing algorithm that guarantees crosstalk bounds are met.

2 Constraint Generation

The most important goal of constraint generation is to maximize the flexibility of the synthesis tool. We propose a methodology for high performance system design which incorporates sensitivity analysis (both digital and analog), intelligent constraint generation, and more powerful constraint-driven layout synthesis. Without significant sensitivity analysis we may burden synthesis with too many constraints and without the capability to handle more general constraints, we may overconstrain synthesis.

2.1 Crosstalk Sensitivity

Typically, crosstalk voltage amplitude is what determines circuit failure by crossing a transistor threshold, which causes a static hazard (as in the case of domino logic), or causing an extra charging delay, which induces a delay hazard. We define node i 's noise voltage bound and its sensitivity to switching on node j .

$$\begin{aligned}\Delta V_i &\leq \Delta V_{B_i} \\ S_{ij} &= \frac{\delta V_i}{\delta V_j}\end{aligned}$$

For digital circuitry, we can normalize for all noise sources j to the common power rail swing: $\Delta V_j = \Delta V$. As we must design the worst case for digital circuitry, we assume all noise contributions are maximally correlated. We assume coupling is linear over the full voltage range. We define crosstalk constraints for node i as

$$N(i) = \sum_j S_{ij} \leq B(i) \quad (1)$$

We factor this sensitivity into analog (A) and digital (D) sensitivities as well as a physical coupling term Q . The analog factor includes the basic circuit interaction, such as relative drive strength. In general, we use circuit simulation techniques to obtain analog sensitivity as described in [3]. The digital factor (D) includes temporal and logical separation of node interaction which we discuss in detail in the next section. Since coupling is primarily through charge sharing, Q is defined by a capacitive divider relation.

$$\begin{aligned}S_{ij} &= A(i, j)D(i, j)Q(i, j) \\ Q(i, j) &= \frac{C(i, j)}{C_{T_i}}\end{aligned}$$

2.1.1 Digital Crosstalk Sensitivity

Crosstalk impacts digital circuits by introducing both logic and delay hazards. For example, asynchronous lines and precharge structures are especially susceptible to crosstalk glitches. Since increasing the cycle time will not alleviate such hazards, these

* This project is supported by Intel Corporation and the Semiconductor Research Corporation under grant 92-DC-008. Its support is gratefully acknowledged.

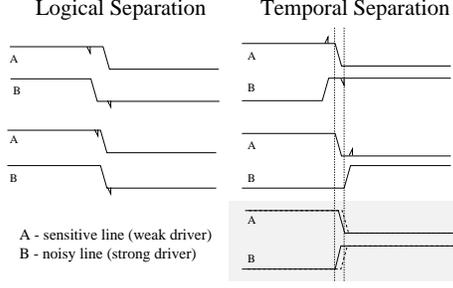


Figure 1: Digital Sensitivity Concept

are static or logical hazards. Coupling between nodes with close timing can induce additional delay by injecting charge on the more weakly driven node, and potentially cause a delay hazard.

Logic and timing are significant correlations which we can use to determine whether physical separation is necessary to decouple two nodes. Some logic faults can only occur with proper timing of the interfering signal (e.g.: a precharged node is sensitive only after the precharge cycle). Also, only nodes which have opposing closely timed transitions can induce a delay fault, as illustrated in Figure 1. Here, signal A and B are not correlated unless they have opposing transitions and switch nearly simultaneously. We term this concept *digital sensitivity*. We propose the following crosstalk constraint generation methodology for digital circuits:

1. Use timing analysis to generate transition and sampling windows which determine timing correlation
2. Use logical transition information to eliminate false interactions. 1) and 2) form the digital sensitivity term D.
3. Apply circuit analysis on node types (e.g.: statically driven versus dynamic node) to produce the analog term A(i,j).
4. During synthesis produce coupling terms Q to satisfy B

By considering only correlated nodes, we can reduce the number of constraints delivered to the layout synthesis stage. By issuing the entire node bound, we allow synthesis tools to perform any necessary partitioning based on physical coupling information. In summary, this approach gives flexibility to synthesis tools to be able to meet absolute bounds. More importantly, digital sensitivity provides us with more information than what nodes interact - it also tells us which nodes can act as relative shields.

3 Crosstalk Channel Routing

As an example of a synthesis tool which supports this paradigm of more powerful constraints and more intelligent synthesis, we describe a new channel routing technique for avoiding crosstalk. Our approach is novel in the sense that the entire node noise bound is handled directly by the router and that routing is performed in such a way as to guarantee those bounds.

3.1 Channel Crosstalk Model

Our model of channel crosstalk focuses on the horizontal shared extent (for simplicity, we assume only a single horizontal wire per node) of two nodes. Taller wires (as processes scale) and relatively small channel widths diminish the importance of contributions from vertical wires. Vertical wiring isolates, and planarized processes reduce, coupling between nonadjacent layers (ie, multi-layer channel routing). So track position assignment, to first order, determines crosstalk between circuit nodes.

In Figure 2, node F is *adjacent* to node E, but not to nodes G or C. Consider sensitive node E with $B(E) = 15$. We have drawn *adjacent* node contributions to E as solid lines. In the FAIL case we note contributions to E of (A:6,D:0,F:7,G:4) yielding $N(E) = 16$. In the PASS case, node A and node C have swapped tracks: (C:4,D:0,F:7,G:3) yields $N(E) = 14$. Thus, crosstalk coupling can be reduced by placing relatively quiet nodes or actual shields between interacting nodes. In either case, we consider the resulting coupling contribution to be negligible.

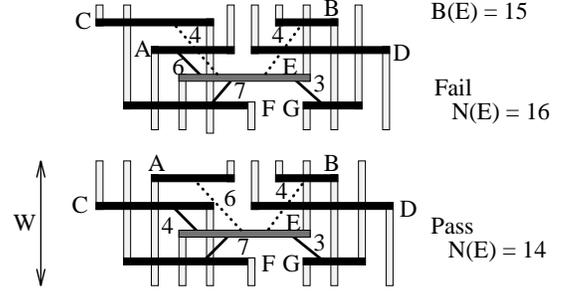


Figure 2: Crosstalk Avoidance Channel Routing

Definition 1 For now, let $pos(i)$ represent relative track position. Let $I(i)$ represent the horizontal interval occupied by node i . Two nodes i and j are *physically adjacent*:

$$PAdj(i, j) = \begin{cases} 1 & \text{if } |pos(j) - pos(i)| < 2 \text{ and } I(i) \cap I(j) \\ 0 & \text{otherwise} \end{cases}$$

Definition 2 A *potential coupling contribution* $\eta(i, j)$ from vertex j to vertex i is defined as follows:

$$q(i, j) = \frac{Q(i, j)}{PAdj(i, j)}$$

$$\eta(i, j) = A(i, j)D(i, j)q(i, j)$$

Thus our channel router constraint on node i is:

$$\sum_j \eta(i, j)PAdj(i, j) \leq B(i) \quad (2)$$

If noise bounds cannot be met using relatively quiet nodes, shields can be added to meet the constraints, potentially increasing the width of the channel. The goal of coupling-constrained channel routing is to assign positions to nodes in the channel satisfying all node noise bounds while minimizing channel width W .

3.2 Track Assignment

We assume a routing region with fixed terminals on rectilinear upper and lower boundaries and floating terminals on left and right boundaries to be routed in an HV or VHV style (one horizontal tracking layer). Our router is general in handling all gridless design rules; we refer to [5] for elaboration of such rules.

There are two types of constraints in the standard channel routing problem: horizontal (interval) and vertical (pin) constraints. The vertical constraints force the horizontal segment of one node to be above the another due to vertical pins which oppose each other, forming a directed graph, which we denote $G_V = (V, E)$. For example, in Figure 2, E must be above G. The horizontal constraints capture which nodes cannot share the same track. These constraints form an interval graph which we denote $G_H = (V, U)$. As introduced in Glitter [5], we solve the track assignment problem by directing the undirected edges of G_H with the objective of minimizing total graph height in G_V .

3.2.1 Basic Track Assignment Algorithm

We illustrate a simplified version of the basic unconstrained track assignment algorithm. We assume in the sequel that $G_V = (V, E)$ is a directed acyclic graph (DAG). Each edge $(i, j) \in E$ has a length $len(i, j)$. Let $FI(v) = \{i : (i, v) \in E\}$ and $FO(v) = \{j : (v, j) \in E\}$. Define vertex *positions* relative to sink and source of G_V :

$$pos(s) = Pos(t) = 0$$

$$pos(i) = \max_{j \in FI(i)} \{len(j, i) + pos(j)\}$$

$$Pos(i) = \max_{j \in FO(i)} \{len(i, j) + Pos(j)\}$$

$$EdgeCost(i, j) = \max \begin{cases} pos(i) + len(i, j) + Pos(j) \\ Pos(i) + len(i, j) + pos(j) \end{cases}$$

The basic track assignment algorithm iteratively directs edges $u \in U$ minimizing impact on total graph height (*EdgeCost*).

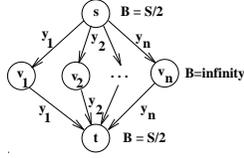


Figure 3: Partition Instance as Coupling Constrained Graph

Track Assignment Algorithm:

```

while (U != 0)
  select edge u by maximum EdgeCost()
  if (pos(i) + Pos(j) < pos(j) + Pos(i))
    insert (i,j) into E
  else
    insert (j,i) into E
  update pos(), Pos()
  delete u

```

In the unconstrained case, $pos(s)$ and $Pos(s)$ can be computed in linear time using a topological sort. As a result, the basic algorithm is $\Theta(V^2E)$. To include crosstalk coupling constraints we require the ability to compute $pos(s)$ and $Pos(s)$ efficiently.

3.3 Coupling-Constrained Levelization

We now consider a simplified subproblem of the graph-based channel routing approach. We show that the constrained levelization problem is NP-complete motivating the use of heuristics in estimating minimum vertex distances to source and sink.

Definition 3 A coupling-constrained graph $G = (V, E, B, \eta)$ is a DAG with source vertex s and sink vertex t , and coupling constraints as in Equation 2.

Definition 4 A position assignment $P : V \rightarrow Z$ of G , is denoted $pos(v), v \in V$, with $pos(s) = 0$. A valid position assignment, denoted \hat{P} , of G is a position assignment satisfying the following topological and coupling constraints:

$$(i, j) \in E \Rightarrow pos(j) - pos(i) \geq len(i, j)$$

$$\sum_{j \in adj(i)} \eta(i, j) PAdj(i, j) \leq B(i)$$

The following problem is of fundamental interest in the theory of performance oriented layout for VLSI circuits.

Definition 5 Coupling Constrained Levelization (CCL): Given a coupling-constrained graph G and a positive integer h , does there exist a valid position assignment \hat{P} of G with $pos(t) \leq h$?

Theorem 3.1 CCL is NP-complete.

Proof:

CCL is in NP: P can be checked for validity in linear time.

CCL is NP-hard: We reduce from the PARTITION problem:

PARTITION [6]: Given a set $Y = \{y_1, \dots, y_n\} \subset Z^+$ where $\sum_i y_i = S$, determine if $\exists W, X \subset Y : W \cup X = Y, W \cap X = \emptyset$ where $\sum_i (y_i \in W) = \sum_i (y_i \in X) = \frac{S}{2}$. We denote $Part(Y)$ as the set of all partitions (W, X) of Y .

Given an instance of the partition problem, construct the coupling constrained graph $G = (V, E, B, \eta)$, as shown in Figure 3.

- For each $y_i \in Y$ create a vertex $v_i \in V$, with $B(v_i) = \infty$.
- Add a source s and sink t with $B(s) = B(t) = \frac{S}{2}$.
- Let $E = \{U_i(s, v_i)\} \cup \{U_i(v_i, t)\}$, $\forall (i, j) \in E, len(i, j) = 1$, and $\eta(s, v_i) = \eta(t, v_i) = y_i$.

Claim: $\exists \hat{P}$ of G with $pos(t) \leq 3$ if and only if $Part(Y) \neq \emptyset$.
Sufficient: Suppose $Part(Y) \neq \emptyset$. Consider an arbitrary partition $(W, X) \in Part(Y)$. Let (ω, χ) be the corresponding vertex partition of V . Define

$$\hat{P} = \begin{cases} pos(s) = 0 \\ pos(t) = 3 \\ \forall v_i \in \omega pos(v_i) = 1 \\ \forall v_j \in \chi pos(v_j) = 2 \end{cases}$$

$$\text{So : } \sum_i \eta(s, v_i) PAdj(s, v_i) = \sum_{v_i \in \omega} \eta(s, v_i) = \sum_{y_i \in W} y_i = \frac{S}{2}$$

A symmetric equation holds for t and so \hat{P} is indeed valid.

Necessary: Suppose $\exists \hat{P}$ of $G : pos(t) \leq 3$.
 $\forall_i (PAdj(s, v_i) + PAdj(t, v_i)) \geq 1$ otherwise $pos(t) > 3$. Let

$$\omega = \{v_i \mid PAdj(s, v_i) = 1\}$$

$$\chi = \{v_i \mid PAdj(t, v_i) = 1\}$$

A vertex is adjacent to either s or t , so $\omega \cup \chi = V$. Thus

$$\sum_{v_i \in \omega} \eta(s, v_i) + \sum_{v_j \in \chi} \eta(t, v_j) \geq S$$

$$\text{But } \hat{P} \text{ is valid : } \sum_{v_i \in \omega} \eta(s, v_i) \leq \frac{S}{2} \text{ and } \sum_{v_j \in \chi} \eta(t, v_j) \leq \frac{S}{2}$$

$$\text{So we have : } \sum_{v_i \in \omega} \eta(s, v_i) = \sum_{v_j \in \chi} \eta(t, v_j) = \frac{S}{2}$$

(ω, χ) are disjoint and correspondingly $(W, X) \in Part(Y)$. ■

3.4 Directed-Path Graphs and \hat{P} -Equivalence

We now look at a special case of the levelization problem where we can find a polynomial solution.

Definition 6 Coupling-constrained graphs $G = (V, E, B, \eta)$ and $G' = (V, E', B', \eta')$ are \hat{P} -equivalent if, for any position assignment P , P is valid for G if and only if P is valid for G' .

Definition 7 G satisfies the directed path property if and only if $\forall i, j \in V$, there is a directed path from i to j or from j to i .

3.4.1 Unconstrained \hat{P} -Equivalent Graph Construction

For graph $G = (V, E, B, \eta)$ satisfying the directed path property, given positioning \hat{P} , every vertex can be adjacent to at most one node above and one node below in a directed path. Construct an unconstrained graph $G' = (V, E', \emptyset, \emptyset)$ as follows:

- $\forall (i, j) \in E : \eta(i, j) > B(i)$ or $\eta(j, i) > B(j)$, define $(i, j) \in E'$ with $len(i, j) = 2$.
- $\forall v \in V, i \in FI(v), o \in FO(v) : \eta(v, i) + \eta(v, o) > B(v)$ define $(i, o) \in E'$ with $len(i, o) = 3$.

Theorem 3.2 Given coupling-constrained graph $G = (V, E, B, \eta)$ satisfying the directed path property, let $G' = (V, E', \emptyset, \emptyset)$ be the graph constructed as above. G and G' are \hat{P} -equivalent.

3.5 Detailed Algorithm

The previous section yields a heuristic solution for levelizing the entire channel routing graph $G_V = (V, E)$: $G_H(V, U)$ is an interval graph and is therefore composed of a linear arrangement of cliques. Clique graphs, when fully directed, satisfy the directed path property. By partitioning our constraints onto the cliques and solving, we can levelize G_V as required in track assignment.

To partition our problem onto the cliques, we need to partition both the crosstalk (edge) contributions and bounds (vertex). We

define the clique subgraph $G^q = (V^q, E^q, U^q)$ corresponding to clique $q \in G_H$. Every vertex $v \in V^q$ maps to a unique vertex $v \in V$, so we don't distinguish their names.

$$(i, j) \in E^q \text{ if } i, j \in V^q \text{ and } (i, j) \in E$$

$$(i, j) \in U^q \text{ if } i, j \in V^q \text{ and } (i, j) \in U$$

Thus, G^q is the projection of *metagraph* G onto clique q where every vertex $v \in V^q$ maps to a unique vertex in V , every $e \in E^q$ maps to a unique edge in E and every undirected edge $u \in U^q$ maps to a unique undirected edge in U .

3.5.1 Crosstalk Partitioning

We partition the crosstalk contributions of $(i, j) \in E$ according to the relative length of the interaction in each clique. We require that bound allocation add up to the original bound:

$$\eta^q(i, j) = \frac{\eta(i, j)}{\|I(i) \cap I(j) \cap I(q)\|}$$

$$\sum_{q \in Q} B^q(i) = B(i)$$

3.5.2 Bound Allocation

We produce $B^q(v)$ to cover clique edge contributions $\eta^q(v, j)$ in order of their cost or impact on channel width. The goal is to allow nodes along critical paths to be physically adjacent.

- $(i, j) \in E$: $pos(i) + len(i, j) + Pos(j)$
- $(i, j) \in U$: $\max \left\{ \begin{array}{l} pos(i) + len(i, j) + Pos(j) \\ Pos(i) + len(i, j) + pos(j) \end{array} \right.$

For vertex v_i , we allocate $B(i)$ to vertices v_j^q to cover their most critical edges, in the following category order

1. critical edges $(i, j) \in E \cup U$
2. critical input/output edge pairs: $i, v, o \in V$
 - $(i, v) \in E, (v, o) \in E \cup U$
 - $(i, v) \in E \cup U, (v, o) \in E$

3. allocate remainder proportional to η^q

Once constraints have been partitioned among all cliques, they are transformed into topological constraints in each clique as described in 3.4.1. Levelization of the G_V is accomplished taking assigning $pos(v) = \max_q pos(v^q)$ and similarly for $Pos(v)$.

Since we operate on clique subgraphs, our algorithm is more complex than the basic algorithm which is $\Theta(V^2 E)$. If we let d be the density of the channel, then we know that the degree of each clique vertex is at most d . So our levelization routines run in $\Theta(E * d * |Q|)$. Thus our overall algorithm is $\Theta(V^3 * E * d)$.

4 Results

We establish a method for measuring the robustness of our constraint generation and routing technique. As there are no standard benchmarks for crosstalk constraints, and no simple measure of the difficulty of a particular constraint set, we show results over a range of constraints. We use standard router benchmarks: 3a, 3b, 3c, 3cr, ddr, ex1, ex2, ex3, ex4, n1, n2, n3, n4, r1, r2, r3, r4, r5 with a total density of 293 tracks, routed nominally in 395 tracks.

We generate constraints over two ranges: increasing signal interaction and increasing signal sensitivity. To model signal interaction or digital sensitivity (D), we assume signals are uniformly distributed in N categories (e.g.: time-windows), varying N as a parameter. We set analog sensitivity (A) equal to unity, and model the noise bound as a percentage of total line-to-line coupling allowed with noisy nodes (ie, in the same signal interaction category). Figure 4 illustrates how total channel width (in percent) varies with $N \in \{1, 2, 4, 6, 8\}$ for different allowed percentages of crosscoupling. One can see that the router hits the worst case of 2x channel width (alternating shields and wires) as expected for $N = 1$, but that digital sensitivity analysis pays dramatically with $N \geq 4$. Our channel router is reasonably efficient in practice¹.

¹For constrained examples, we average 30s of CPU per example on a Intel Pentium-60 running the Linux Operating System

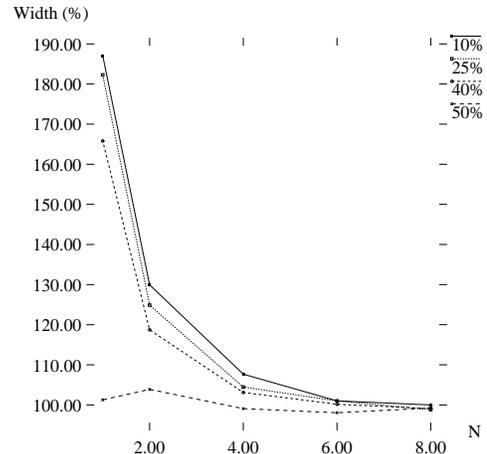


Figure 4: Channel Width vs. Node Interaction

5 Conclusions

We have described a constraint-driven methodology for handling crosstalk as a second-order analog effect in high-performance digital system design, incorporating digital sensitivity analysis as a way to reduce the number of constraints. We have also demonstrated that more general constraints can be handled in layout synthesis, which in turn provide more flexibility to achieve a dense result. We argue strongly that a constraint-driven approach is necessary in digital design as efforts to reduce second-order effects or optimize them do not directly attack the problem, one of correctness. As such, we believe efforts to build synthesis tools, which are correct by construction and optimize area subject to analog constraints, such as our crosstalk-constrained channel router, will be a requirement to build future high-performance systems.

Acknowledgements

Thanks to Adnan Aziz for inspiring the idea of using a PARTITION reduction in Theorem 3.1 and to Chris Lennard for several careful readings and critical discussions.

References

- [1] T. Gao and C. L. Liu. Minimum crosstalk channel routing. In *Digest of Technical Papers of the 1993 IEEE International Conference on Computer-Aided Design*, 1993.
- [2] K. Chaudhary, A. Onozawa, and E. S. Kuh. A spacing algorithm for performance enhancement and cross-talk reduction. In *Digest of Technical Papers of the 1993 IEEE International Conference on Computer-Aided Design*, 1993.
- [3] Henry Chang, Alberto L. Sangiovanni-Vincentelli, Felice Balarin, et al. A top-down, constraint-driven design methodology for analog integrated circuits. In *Proceedings of the IEEE 1992 Custom Integrated Circuits Conference, New York, NY, USA*, pages 8.4/1–6, 1992.
- [4] U. Choudhury and A. Sangiovanni-Vincentelli. Constraint-based channel routing for analog and mixed analog/digital circuits. In *Digest of Technical Papers of the 1990 IEEE International Conference on Computer-Aided Design*, 1990.
- [5] H. H. Chen and E. S. Kuh. Glitter: A gridless variable-width channel router. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 5(4):459–465, October 1986.
- [6] M. R. Garey and D. S. Johnson. *Computers and Intractability*. W. H. Freeman and Co., 1979.