Automatic Protoboard Layout

by

Michael Mekonnen

B.S. EECS, Massachusetts Institute of Technology (2013)
B.S. Mathematics, Massachusetts Institute of Technology (2013)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Masters of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2014

Author . . . .
Department of Electrical Engineering and Computer Science
September 19, 2013

Certified by . . . .
Dennis M. Freeman
Professor
Thesis Supervisor

Accepted by . . . .
Professor Albert R. Meyer
Chairman, Department Committee on Graduate Theses

Automatic Protoboard Layout

by

Michael Mekonnen

Submitted to the Department of Electrical Engineering and Computer Science
on September 19, 2013, in partial fulfillment of the
requirements for the degree of
Masters of Engineering in Electrical Engineering and Computer Science

Abstract

Abstract.

Thesis Supervisor: Dennis M. Freeman
Title: *Professor*

Acknowledgments

Acknowledgments.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Statement

What problem are we solving?

1.2 Motivation

Why is this an interesting problem?

1.3 Goal

Precisely state the goal of this project. In particular, explain that we ultimately want to make a teaching tool for 6.01.

1.4 Outline

How is the Thesis organized?

Chapter 2

Background

2.1 Technical Background

What is a circuit schematic? What is a protoboard? What circuit components are we working with in this project?

2.2 Previous Work

2.2.1 Current tools in 6.01

Discuss CMax and its capabilities.

2.2.2 Current work in automatic protoboard layout

What similar work has been done before?

Chapter 3

Evaluation

How are we going to evaluate a particular solution to the problem?

Chapter 4

Methods

In this Section, I discuss my solution to the problem and various alternatives I considered along the way.

4.1 Overview

I solved this problem by formulating it as a search problem. By this I mean, given a schematic of a circuit, I start from an empty protoboard, and I consider the space of all possible protoboard layouts to find the protoboard corresponding to the schematic at hand. The space of all possible protoboards is very large (?), so I utilize various heuristics to facilitate the search.

I broke down the problem into two parts. The first task is finding a placement of all the circuit pieces on the protoboard. The second task is wiring them up appropriately.

4.2 Part 1: Piece Placement

Let us first consider how to place a set of circuit pieces on the protoboard for a given circuit schematic.

Figure 4-1: Placement of the pieces on the protoboard.

4.2.1 The Pieces

Any given circuit may contain resistors, Op Amps, pots, motors, head connector parts, or robot connector parts. For each of these components, we must put down a corresponding piece on the circuit.

Resistors

For the sake of simplicity, and to significantly reduce the search space (?), for every resistor in the schematic, I use one resistor piece on the protoboard placed in the middle strip of the protoboard as shown in Figure 4.2.1. This choice, i.e. allowing the resistor pieces to only reside in the middle strip of the protoboard, is critical as the resistor pieces can generally be placed at numerous places on the protoboard. With this restriction, there are 63 slots available for one resistor. Without this restriction, there are a total of 763 slots available. The restirction is good when we consider the reduction in the search space size. On the other hand, the restriction is bad when we consider the size of circuits the algorithm can layout. Given that the number of resistors in a typical 6.01 (?) circuit is very small, this restriction proves to be very useful, but we will consider the alternative in Section 4.4.1.

n	$f(n)$
1	1
2	1
3	7
4	25
5	81
6	331
7	1303
8	5937
9	26785
10	133651

Table 4.1: Number of ways of packaging together n Op Amps for various values of n.

Op Amps

Op Amps are the trickiest components to handle because each Op Amp package put on the protoboard contains two Op Amps within it. Thus, we face the task of packaging the Op Amps in the schematic in the "best" possible way, i.e. so as to require as little work as possible when wiring the pieces together. Equation 4.1 presents an expression for the value $f(n)$, the number of different ways to package together n Op Amps. To get a sense of how many different packagings are possible, Table 4.2.1 gives the values of $f(n)$ for various n.

$$f(n) = \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} \frac{n!}{n!(n-2k)!} \tag{4.1}$$

Each Op Amp package is placed in the middle strip of the protoboard, as shown in Figure 4.2.1.

Pots

Each pot piece can be placed in one of two vertical locations on the protoboard. Figure 4.2.1 provides an example of both options.

Head, Motor, and Robot Connectors

We use a 6-pin connector to connect to a motor, and an 8-pin connector to connect to either a head or a robot. Each connector can be placed in one of two vertical locations on the protoboard, as shown in Figure 4.2.1.

4.2.2 Choosing a Placement

When choosing a placement of circuit pieces on the protoboard, we have at hand a plethora of options: each piece can be placed in one of very many places on the protoboard; each piece has two possible orientations; there are numerous ways of packaging together Op Amps; etc.

Simplifications

I reduce this large number of options by only allowing placements in which no two pieces share a column. Once again, this is not necessary in general, but the number of pieces necessary for a typical 6.01 circuit would certainly fit in this framework.

Next, I specify that there be exactly two columns on the protoboard separating each consecuitive pair of pieces, unless the pieces are both resistors, in which case there must be exactly one column separating them. These numbers of columns were chosen to leave enough space for wiring. Given a set of pieces to be put on the protoboard, this specification reduces the problem of choosing a placement for the pieces to finding an *order* of the pieces together with choosing their respective vertical locations and orientations.

Given these simplifications, we have various options as to how to pick a placement.

Random Placement

One simple alternative may be to choose a placement randomly. That is, we choose an Op Amp packaging randomly; we choose an order of the pieces randomly; and we choose the vertical locations and orientations of the pieces randomly as well. The advantage of this approach is that it gives us a placement very quickly without requiring much computation. On the other hand, we may end up placing two pieces that need to be connected to each other very far apart, and we will have a difficult time doing the wiring. Hence, we ought to consider alternatives in which we take into account the task of wiring. We should try to place the pieces so as to require as little work during wiring as possible.

Minimal Heuristic Cost

The key idea is that if two pieces are meant to be connected together by wires, then they ought to be placed close to each other on the protoboard. We can capture this idea by assigning heuristic costs to the placements and choosing a placement that produces the minimal heuristic cost.

Let us first devise the cost function to achieve this goal. Given a circuit schematic and a corresponding placement of the circuit pieces on the protoboard, what do we need to connect with wires? Well, every pair of components in the schematic that are connected by a wire gives us a corresponding pair of locations on the protoboard that ought to be connected by wires. However, we can express this requirement a little bit more concisely. We ought to consider all of the nodes in the schematic, and find the circuit components in the schematic that are connected to the respective nodes. Now for each node in the circuit, we get a set of locations on the protoboard that ought to be interconnected. The first step in devising the cost function we are looking for is to have a way to estimate the cost of connecting two locations on the protoboard. A simple such cost function that comes to mind is the Manhattan distance between the two locations. Recall that we want to produce aesthetically pleasing protoboard layouts, and one of the requirements in achieving this goal is only using horizontal and vertical wires (i.e. no diagonal wires) so the Manhattan distance cost is appropriate. Given this heuristic cost for connecting two locations with wires, we can define the heuristic cost for interconnecting the locations associated with a particular node to be the weight of the minimum spanning tree of the locations. Now we can define the cost of a placement to be the sum over all nodes in the circuit of the cost for interconnecting the locations for each node.

Now that we have a cost function for placements, we can aim to find a placement with the minimal cost. However, this involves trying all possible orderings of the pieces with which we are working. For example, if we are trying to order 10 pieces, we would need to look at $10! = 3628800$ possible orderings. Note that this is in addition to searching over all possible ways of packaging the Op Amps together. It is clear to see that the search for a minimal cost placement quickly gets out of hand. So we aim to find a placement that has a very small, though maybe not minimal, cost.

Small Heuristic Cost

Algorithm 1 presents a procedure that orders a given list of pieces in a way that results in a small cost. The algorithm relies on two ideas. First, once a piece has been placed, all the pieces that are connected to it will be placed soon after so that it is more likely that those pieces are placed close to it. Second, we place the pieces with the most nodes first since those are the one that most likely have connections with many other pieces.

Algorithm 1: Producing a circuit piece placement with small heuristic cost.

```
Data: A list P of circuit pieces.
Result: A list R of circuit pieces representing a placement.
Sort P by number of nodes on the respective pieces;
q ← empty Queue;
R ← empty List;
while P is not empty do
    Pop the first piece in P and push it onto Q;
    while Q is not empty do
        p ← Q.pop();
        Consider all vertical locations and orientations of p;
        Place p at an index in P that minimizes the cost of P;
        foreach piece q in P connected to p do
            Push q onto Q;
```

Using one of the above methods, we can find a placement of circuit pieces on a protoboard. Our next task is wiring them together to produce a circuit equivalent to the circuit schematic of interest.

4.3 Part 2: Wiring

4.3.1 What do we need to wire together?

4.3.2 Search Infrastructure

4.3.3 Wire all pairs at once

4.3.4 Wire one pair at a time

4.4 Why breakdown problem into two parts?

4.4.1 Treating Resistors as Wires

Chapter 5

Results

Quantitatively compare the various methods discussed in the previous section.

Chapter 6

Discussion

6.1 Explaining the Results

Give plausible explanation for the observed results.

6.2 Remarks

Why are these results encouraging? What are their implications? Relate back to Introduction to Thesis. What could have been done differently?

Appendix A

Schematic Drawing GUI

Discuss the features and capabilities of the GUI.