ETM

ENGINEERING & TECHNOLOGY MANAGEMENT

Title: Introducing Printed Circuit Boards with Optimal Revision Costs: A Linear Programming Model

Course:

Year: 1989

Author(s): A. Doumani, R. Phillips, R. Ritter and B. Theus

Report No: P89003

Abstract: We utilize a Linear Programming model to help a project manager decide on the optimum time for the revision of an existing circuit board, and how to minimize tooling and part costs. Minimizing costs is the objective of the report; determining the optimum time to initiate the changes is the key to achieving that objective

INTRODUCING PRINTED CIRCUIT BOARDS WITH OPTIMAL REVISION COSTS

A. Doumani, R. Phillips, R. Ritter,
and B. Theus

EMP - P8903

EMP - P8903

EAS 543
June 5, 1989
Dr. Kocaoglu

Introducing printed circuit boards with optimal revision costs:

An LP model with design curves, late penalties, and resource constraints.

Good approach to use L.P. in an ingenious way to solve a non-linear problem

Alex Doumani
Robyn Phillips
Russ Ritter
Barbara Theus

EXECUTIVE SUMMARY

This project utilizes a mixed model linear program to help a project manager decide when the optimum time is to revise an existing circuit board but also minimize the tooling and part costs. Although minimizing costs it the goal of the project, determining the optimum time to initiate changes is the key to minimizing costs. The basic model is solved for Cost 1, Demand 1 and with no late penalty for one solution. Next, a sensitivity study is completed to see the impact of changing the cost, demand, and late penalty. The sensitivity study indicated that the demand variations and late penalties to a lesser extent tooling and part cost variations have the most significant impact on the optimum time to revise a board and at what cost.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

INTRODUCTION

A manager's decision of when to introduce a new product or improve an existing one is critical. The timing of this decision is especially important in the fast paced electronics industry. Managers need an effective aid to help them determine when a prototype is stable enough to send into production, and what the impact of mis-judging the "right" time is. Penalties for introducing a product late into a market are high in an industry where it is critical to be the first out with something new. Our project utilizes linear programming to help decide when to initiate the changes required for introducing computer electronics. Minimizing the total product and tooling costs during the time considered is this project's goal, but finding the optimum time for a new revision is the key item of interest. The costs which are being optimized are related to the different revisions of a printed circuit board, where the assembly costs for each revision differs.

Common factors involved in introducing a new board include tooling costs and tooling resources, which we defined as the set-up cost including the required lead time for a revision change and other discrete costs. The cost of making the changes from one revision to the next must be weighed against the benefits, while also considering the future demand. Limited resources and the costs of revising multi-layer circuit boards are studied during the pre-production phase.

We examined the trade-offs between changing the board versus paying an incremental cost to modify an "older revision" circuit board so it would be functionally equivalent. Building the boards later as a way to reduce the rework costs, or after more of the defects are discovered is allowed, but within specific limits. A penalty for project delay is used to study the trade-off for delaying the project inn an effort to reduce the unit costs versus incurring the late penalty.

To develop realistic cost constants, we assumed the debugging effort for a new product is comparable to previous products, or that a similar product within the industry might show a typical debugging "learning curve". We created from these non-linear learning curve assumptions a set of discrete unit costs showing lower costs as the design becomes stable, which means fewer changes required. We then used a linear decision model to demonstrate how cost trade-offs show the best time to introduce a new revision. The model's objective is to minimize the cost of a design revision by evaluating different learning curve "starting points" and rate-of-change assumptions. In addition, we also tried different sets of demands and late penalties.

Good approach

Post-optimal analysis of the cost constants, including the late penalties, tooling costs, and the demand forecasts complete the sensitivity analysis. We discuss the results of the various model versions on the decisions, and how these might be interpreted by the project manager. The sensitivity analysis allows us to examine how changing the costs, demand, and late penalty affect the decision of when to initiate a new revision. We mechanically modified the "learning curve" cost in the final time cycles, and in some cases we took it to zero. This is equivalent to a project in which all design defects are found and fixed prior to the product being marketed. Our conclusion is also directed to the project manager and how judicial application of this simplified model might help reduce product introduction costs.

LITERATURE SEARCH

One key item we assumed in this model is that the design debugging effort has a degree of "predictability" which a project manager may use to make decisions about new revision timing. This predictability may come in the form of "learning curves" which are often applied to high volume, repetitive industries. Branam demonstrated the Learning Curve Theory as applied to a new product introduced in a job shop, or "engineered-to-order" environment. (1) In place of the unit cost improvement as related to the learned production efficiency portrayed in Branam's research, we developed unit costs relative to an incremental design improvement.

ok

Historical data from the company we studied, as well as data about design debugging and reliability growth during prototyping in the electronics industry support our proposal that design stability can be represented by a learning curve. (2) (3) As Branam said, "What is important about this (learning curve) theory is that the rate of improvement is regular enough to be predictable." (4)

One caution to note about our model relative to the "learning curve", is that our curve was not based on the number of units tested. Rather, it was simplified to show improvement over time, which does not demonstrate the impact of a project manager accelerating the design stability or reducing costs by increasing the debug resources. These resources would include the number of engineers, prototype units, and test equipment. For the purpose of many developments where cumulative "test" experience is at a relatively steady rate and the resource budgets are essentially fixed, simplifying the earlier time cycles should not distort the model significantly.

Delaying the project to reduce costs is another factor we apply in our model. Smith-Daniels and Aquilano demonstrated how late-start project scheduling can improve the net present value of a project, but their results were gained through the effects of the time-value of money in project cash flow. (5) They demonstrated an average 2.5% net present value improvement by reducing task "slack" to delay cash payments during the projects's life. This assumes the projects are paid for at the "end" as the product goes to market. We looked at the impact of delaying the project to affect cash flow, but not based on the "time value of money". Our interest is more in the "money value of time" as it relates to delaying the introduction of a product to reduce the costs associated with the design stability/instability.

The cost to switch from one "model" or revision to another must be accompanied by a "set-up" or tooling cost. Frendewey and Sumchrast showed us a much more robust method of modeling set-ups, with constraints for set-up lead times and labor; including labor constraints and penalties for overtime. (6) We applied these constraints to our electronics effort, where some of the labor used to revise the board is also used in the project design debugging effort. However, we greatly simplified our model by combining these cost and constraints into fewer variables. In a project where tooling is primarily done by an outside vendor, our discrete cost approach provides a reasonable and manageable model.

PROBLEM FORMULATION

As stated earlier, this problem is formulated as a single objective, mixed-model, linear programming model, with set-up costs and resource limitations.

Mixed-models

The "mixed models" are the various revision levels of the printed circuit assembly. The unit "modification" cost for the printed circuit which relate to design changes can be described as wires added, and changes made to the components or parts soldered to the board. By determining the cumulative number of wires required for a new revision or "fab model" when the tooling occurred we can find the revision costs. The printed circuit assembly model used to satisfy a demand in time 4 (Ti, i=4) with revision B artwork would use the nomenclature T4B. These are the primary apparent decision variables in our linear model. We say " apparent", because the decision to meet a period's demand with a certain number of Rev. B instead of A is really a decision forcing the change with adequate lead time so that B may be built.

Time increment:

The time increment for the model has been defined as half the lead time from initiating a new revision until it is available for consumption. For example, a set up started in time 2 (T2) will result in the new revision being available two periods later or time 4 (T4). Given our target study, each time period represents about four weeks, except as noted in the model, where the last two periods are production quarters. The model could just as easily represent time scaled to industries with longer or shorter lead times.

Model Unit Costs:

The material replacement or repair cost per assembly declines in each time period as the design debug progresses without affecting the lead time or tooling costs. On the other hand, the design stability increases for each time period, and the costs associated with future "revisions" can be estimated. This simplification eliminates the non-linear functions associated with the prototype "learning curve" and design stabilization effort.

The following table represents a sample of the information required to calculate these costs for Cost Model 1, Demand 1. The data shown here is based on a theoretical learning curve of about .8 in the early cycles and with a starting point at $1,000 of design defects still to be discovered. The defects may be either wires or parts.

TABLE 1 Design Changes Affecting Unit Costs

# OF WIRES (W)	REV. (Rj)	TIME PERIOD (Ti)	# OF COMPONENTS (P)
100	A	T1	50
80	B	T2	40
65	C	T3	38
50	D	T4	25
40	E	T5	20
20	F	T6	10
5	G	T7	2
0	H	T8	0

The wires cost \$5 each, while the components are \$10 each. With this information we have the following equations:

$$Cr = \$5 \times W \quad \text{and} \quad Ct = \$10 \times P$$

where Cr = cost of adding wires and Ct = cost of component changes. These two variables are used in the next equation.

$Cj = \text{Unit Cost} = [Ct(Ti)] + [Cr(Rj)]$

For example, the unit "design change" cost of a Rev. B board built in time period 7 (T7) would be found as follows:

$Cr7 = \$5 \times 80 = \$400$

$Ct7 = \$10 \times 2 = \$20$

$C7 = Ct7 + Cr7 = 20 + 400 = \$420$

Graphically, the rework costs associated with the cumulative design flaws "yet to be discovered" appears as follows:

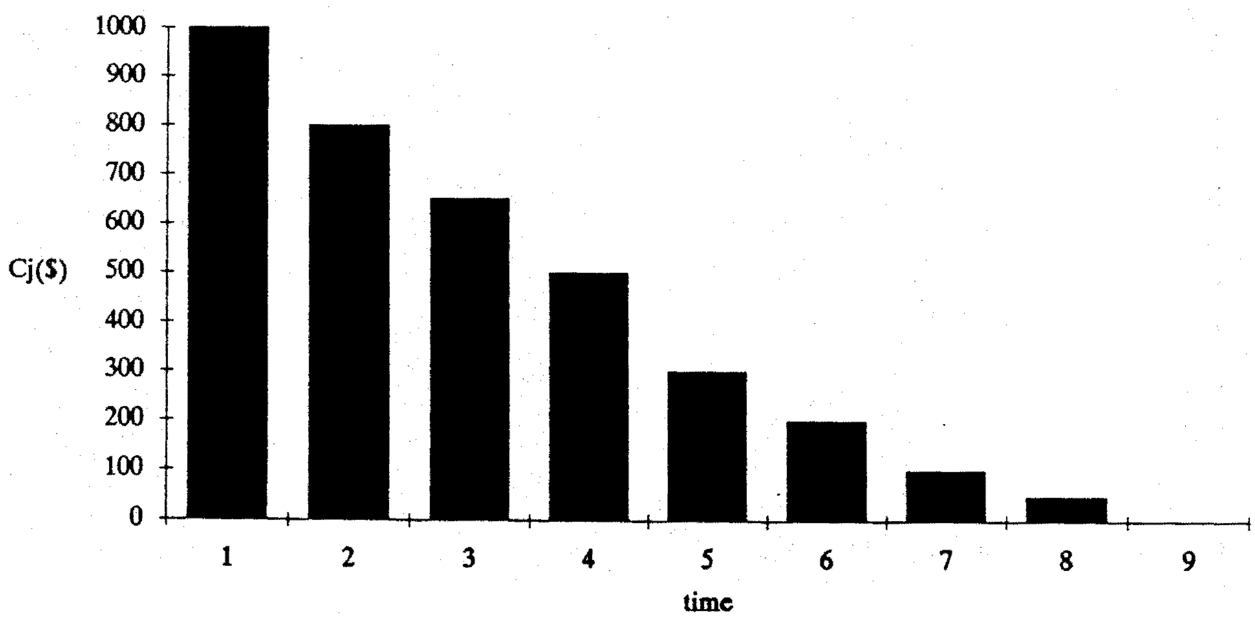

FIGURE 1 Cost Model 1

The comparison against the tooling costs is also shown.

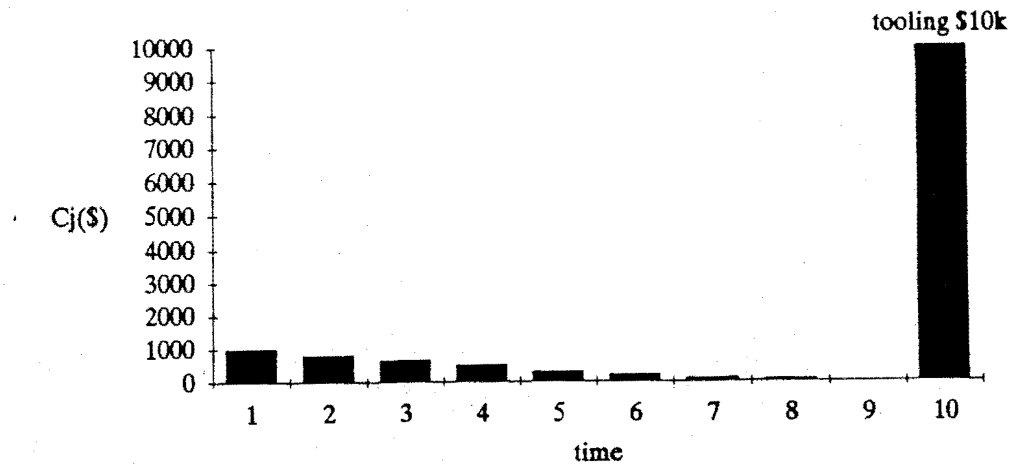

FIGURE 2 Cost Model 1 compared to Tooling Costs

The model objective function is:

Minimize: SUM[Cj(Xi)] + SUM[Sj(Revj)] + LP

where:

Cj = linearized design-change costs as described earlier.

Xi = TiRj = the number of boards built in time T using fab revision Rj (T5B)

Sj = cost of set up "fab artwork re-tooling" occurring in time j (j= i - 2) (eg. $10,000)

Revj = Set up incurred for a revision of artwork in time i - 2 (eg. C)

LP = late penalty if Q1 demand is delayed by a time period (eg $500,000)

The model was set up with the change of revision levels (Revj) and late Penalties being integer (0/1) variables.

Resource constraints:

As usual, there are many constraints which must be considered, therefore, they are each detailed below.

Production demand is a constant and each period's demand must be satisfied. The demand must be met in time T_i or in time $i + 1$, which is a delayed or "late" period for b_i. Because of the Go/No-Go constraints we used demand as a coefficient in the model and had to change them manually. B_i is the forecast of pre-production units :

$$\text{SUM}\ (T_iR_j + (T_i + 1)R_j) >= b_i$$

Demand 1 is shown in table 2.

Table 2 Demand 1

```
+--------------------------------------+
|                                      |
|       Time             Demand        |
|                                      |
|        1     -----+                  |
|                   +------   10       |
|        2     -----+                  |
|                                      |
|        3     -----+                  |
|                   +------   20       |
|        4     -----+                  |
|                                      |
|        5     -----+                  |
|                   +------  100       |
|        6     -----+                  |
|                                      |
|        7     -----+                  |
|                   +------  100       |
|        8     -----+                  |
|                                      |
|        9     ------------  900       |
|                                      |
|                                      |
+--------------------------------------+
```

At this time a few definitions of phases within the development cycle are required. Alpha is the internal "real application" test phase; beta is the external "real application" test phase where a non-paying customer uses the product. Strife is early customer purchases with a low demand.

The boards made during time period 1 and 2 are "alpha" parts, while 3 and 4 are "beta". Strife occurs during periods 5 and 6. Next, is the production during quarter 1 (Q1) and finally quarters 2 through 4. These last two divisions are production phases.

We limited the number of total tooling revisions in the development cycle to 2, which recognizes limited project engineering resources. Because of this constraint, there may be instances when the unit assembly cost multiplied by the demand would otherwise offset the revision cost, but we are unable to turn the board. No similar rigid bound exists to require "clean" artwork, which allows the option of never revising if the assembly cost does not offset the tooling cost within a selected "demand" period. Therefore, the total number of revisions is defined as:

$$\mathrm{SUM}\ (\mathrm{REV}j) <= 2$$

No revision of a board except A is allowed to be produced without a "set-up" for that revision.

$$\mathrm{SUM}\ (-\mathrm{TiR}j) + bi(\mathrm{REV}j) >= 0$$

The late start of an alpha or beta board is allowed, but if alpha starts late, beta must start late which dictates the next constraint equation:

$$aij\ (Ti - 1Rj) + \mathrm{SUM}(Xi) >= 0$$

where Ti - 1 represents a delayed build or "late" decisions in the prior time period. For example, period 2 for alpha and period 4 for beta. Xi is limited to "on time " X for periods 1, 3, 5, and 7 in the current period and where aij = -b(i + 1).

From above, if beta starts late, Q1 production starts late and incurs a late Penalty. The late penalty is forced into the decision by:

SUM(-T8Rj) + b4(LP) >= 0

In this model b3 or "strife" demand is disconnected from the late penalty. This reflects an industry where strife or accelerated life testing occurring prior to production does not influence the ship date unless fatal flaws are uncovered.

The model for Cost 1, Demand 1, High Late Penalty is shown on Table 3.

TABLE 3 COST MODEL 1 MATRIX

			late			late						late(no penalty)											
	Var	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
LP model: nam		T1A	T2A	T3A	T3B	T4A	T4B	T4C	T5A	T5B	T5C	T5D	T6A	T6B	T6C	T6D	T6E	T7A	T7B	T7C	T7D	T7E	T
Objective: Ai -		1000	900	825	725	750	650	575	650	550	475	400	600	500	425	350	250	550	450	375	300	200	1
Constraints:																							
Line 1		1	1																				
2				1	1	1	1	1															
3									1	1	1	1	1	1	1	1	1						
4																		1	1	1	1	1	
5																							
6					-1		-1																
7								-1															
8										-1				-1									
9											-1				-1								
10												-1				-1							
11																	-1						
12																			-1				
13																				-1			
14																					-1		
15																						-1	
16																							
17																							
18																							
19																							
20																							
21																							
22																							
23																							
24																							
25			-2.1	1	1																		
26						-5.05	-5.05	-5.1										1	1	1	1	1	
27																							
28																							

		late																							
21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44		
E	T7F	T8A	T8B	T8C	T8D	T8E	T8F	T8G	T9A	T9B	T9C	T9D	T9E	T9F	T9G	T9H	LP	B	C	D	E	F	G		
0	150	525	425	350	275	175	125	50	500	400	325	250	150	100	50	50	500000	10000	10000	10000	10000	10000	10000	Minimize	
																									b
																								>	10
																								>	20
																								>	100
	1	1	1	1	1	1	1	1																>	100
									1	1	1	1	1	1	1	1								>	900
																		20						>	0
																			20					>	0
																		100						>	0
																			100					>	0
																				100				>	0
																					100			>	0
			-1															100						>	0
				-1															100					>	0
					-1															100				>	0
						-1															100			>	0
	-1						-1															100		>	0
								-1															100	>	0
										-1								100						>	0
											-1								100					>	0
												-1								100				>	0
													-1								100			>	0
														-1								100		>	0
								-1															100	>	0
															-1	-1							900	>	0
																								>	0
	1																							>	0
		-1	-1	-1	-1	-1	-1	-1									100							>	0
																		1	1	1	1	1	1	<	2

It is appropriate to go over Cost Model 1, Demand 1, with no late penalty, in detail.

LINE #

1 - Alpha demand
2 - Beta demand
3 - Strife demand
4 - Quarter 1 demand
5 - 6 - Quarters 2 - 4 demand
7 - 23 - Constraints limiting the use of a revision without first turning the artwork.
24 - If alpha is late, beta is late
25 - If beta is late, Q1 is late
26 - Late Penalty for late production
27 - Limits the number of revisions to 2.

SOLUTION

Our basic model using Cost 1, Demand 1, and 0 Late Penalty, had the following results:

Rev. to Turn	Time to Turn	Obj. Funct.
E G	4 6	$119,000

This information tells us that we should change to Revision E in time period 4. The new revision will be available in time 6 (T6). In time period 6 (T6) we should initiate changes to make Rev. G, which will be ready in T8. The cost of the tooling and parts will be $119.000.

SENSITIVITY ANALYSIS

The dependencies of bi within the model forced a partial change of the model costs and demands built into the model. Changing just bi would not have been meaningful because of the interdependences. The standard LINDO Range command was not useful in our case as stated earlier because the right hand side of the equations are related to the coefficients in the constraints. Also, the costs are related to the cost formula and as a result are related to each other. Thus, when doing the sensitivity study, we had to manually change some of the coefficients. As a result, we chose three costs, three demands, and three different late penalties which resulted in 27 different models. (See Appendix A) So, we really created a matrix of varying cost, demand, and late penalties. The results of the solutions for cost, demand, and late penalty variations as outlined previously are summarized in Table 4.

Table 4 Model results

Cost (Cj)	Demand (bi)	Late Penalty* (LP)	Rev. To Turn	Time To Turn	Obj. Funct.
1	1	0	E G	4 6	119,000
1	1	.03	E G	4 6	136,500
1	1	.5	E G	4 6	136,500
1	2	0	E G	4 6	515,000
1	2	.03	E G	4 6	545,000
1	2	.5	E G LP	4 6	602,500
1	3	0	G	6	14,850
1	3	.03	NO TURNS		16,600
1	3	.5	NO TURNS		16,600
2	1	0	E G	4 6	327,500
2	1	.03	E G	4 6	357,500
2	1	.5	E F	4 5	447,500
2	2	0	E G	4 6	1,557,500
2	2	.03	E G LP	4 6	1,587,500
2	2	.5	E G LP	4 6	2,057,500
2	3	0	C G	2 6	36,925
2	3	.03	F	5	46,925
2	3	.5	F	5	46,925
3	1	0	E	4	967,500
3	1	.03	C E LP	2 4	997,500
3	1	.5	E	4	1,030,000
3	2	0	C E	2 4	4,037,500
3	2	.03	C E LP	2 4	4,067,500
3	2	.5	C E LP	2 4	4,537,500
3	3	0	E	4	145,100
3	3	.03	F	5	146,850
3	3	.5	F	5	146,850

* NOTE: The late penalty is in millions of dollars.

We need to look in detail at each item we changed to see the effect. First, were the cost coefficients. All three cost models started with an initial cost of $1,000. (See Appendix B) Cost Model 1 is from a medium learning curve, while the second cost model is a fast learning curve.

Finally, Cost Model 3, is a slow learning curve.

As time passes and we move down learning curve 1, the costs do not decrease significantly from the same revision being built in consecutive time periods. Also for Cost Model 1, the improvement in the costs between revisions is small. For Cost Model 2, however, as we progress in time down the learning curve, there is a high improvement in cost for the same model and the change between revisions is also high. The results of changing from Cost Model 1 to Cost Model 2 have no effect on when to change to a new revision.

Finally, for the third model, the difference in the cost of a model built in a later period versus the current cost again only decreases slightly, but the cost improvement between revisions is again high. Changing to Cost Model 3 generally brought both turns earlier, assuming that large improvements in revision levels without a fairly flat learning curve caused this.

Next, we examined the effects of changing the late penalties. For all three cost models this change did not significantly affect when the board should be revised. Although the late penalty did not affect the results, a review of what each late penalty implies is needed. A late penalty of 0 assumes that a company has no competition and will get the full market share even if the product is late. The second late penalty, $30,000, indicates that if a product is late the company will have to lose a very small amount of business. Finally, the third late penalty of $500,000 represents a loss of the market share that your competitors will pick up.

The final item to be examined is what changing the demand did to the results. The second and third demands are detailed in Appendix B. The first demand, or standard demand, for the model is for a moderate volume of boards representative of the "mainframe" business. The second demand is for a demand five times higher, similar to the personal computer industry, while the third demand represents very low volume such as the aircraft engine business. In Cost Model 1 changing the demand had no impact on the time to turn the revision, but in Cost Models 2 and 3 the demand change tended to make the revision times slightly earlier. This indicates that when the variations in the cost of the board is high the demand changes are more significant.

In reviewing the cost variations, it is worth mentioning that Cost Models 1 and 2 are typical in the electronics industry, where there is not a large variation. The third set of costs were artificially inflated to show the impact on the model. This allowed us to look at the validity of the model. If it had been left with only small changes in the costs, we may have reached false general conclusions if the model had low sensitivity. This makes changing the costs over a large range very important.

DISCUSSION OF RESULTS

Model Results

In order to asses our model's validity, we must ask two questions:

1. Do the decisions from the model correspond with reality and offer sensible results?

2. Does the model help a decision maker more than existing trivial decision algorithms?

Due to the use of integer programming for some key variables, for example tooling revision, required a "brute force" examination. As stated earlier, this was also caused because the constraints used in the equations were interrelated and could not be changed independently without affecting other constants. We sampled three cost models, three demand models, and three late penalties. In the end, this resulted in 27 variations of the model.

Based on the 27 primary versions of the model, it appears to track cost, demand, and the penalty assumptions with appropriate results. The only two (2) cases from all of them where a tooling revision is not indicated is when the volume is very low or using Demand Model 3. When demand includes medium to high volume, the model tells us to revise the board at least once, early in the model when the volume or demand is high enough to offset the tooling costs with pre-production rework costs and also typically at the end of a design cycle. In 13 of the 27 cases, this was in Rev G. The end of a design cycle coincides with an early period in the pre-production demand while the design is stable. In Cost Model 2, specifically, the incremental rework costs are driven to zero quickly.

Comparing against "trivial" solution algorithms indicates the model can be used to save project costs. Given resource constraints of two tooling revisions, one decision rule would be "never allow project delay, and revise the artwork after beta or Rev. E", and again before production or Rev. G. Allowing for common trivial checkpoint rules such that there are no wires to eliminate after the "beta" stage: do not revise anything at G. This trivial approach results in the same solution as our model in 13 of the 27 cases. The average difference in the objective functions obtained between our model and the trivial decision algorithm is about 20%. In those seven cases where our model indicated reducing the unit costs would be worth accepting a late penalty, the average difference in tool rework costs between the trivial and modeled solution was 50%. (See Appendix D) Since the costs of revising and expedited prototypes can total over 20% of the development costs in the electronics industry, a 50% savings in the rework portion of the costs is significant.

This represents 10% of the total project development costs. A few non-standard variations of the model were also studied. Eliminating the constraint of only two tooling revisions reduced the rework costs 22% in the "high" cost models sampled. (See Appendix D)

Sensitivity Results

Because of the way the model was constructed, many interdependencies were built into the cost coefficients and the bi. As a result, it was not possible to change one without affecting the others. This made the sensitivity analysis a "brute force" study. Changing the costs into Cost Model 3 had significant impact on the revision chosen for turning. Changing the demand in Cost Models 2 and 3 also tended to make the optimum time of a revision change occur slightly earlier. The late penalty changes did not have an impact on the revision chosen to turn, but it did significantly affect what revision is built in later time periods. From this we know if the variations in the board costs are high, changing the demand affects the model more significantly. Therefore, we might conclude that demand is the dominating factor in this model.

CONCLUSIONS

The model we developed is "brute force". Given that, it has been used to assist in "rule of thumb" project decisions, and in general appears to allow for better decisions than existing trivial algorithms.

Our sensitivity study showed that the three items we changed: costs, demands, and late penalties, each affected the model differently. The cost changes illustrated the relationship between the cost of a model built at different times and the cost of an older model versus a newer one, but did not significantly impact the results. The late penalties and the demands did impact the model's solution. Demand became quite important when the board cost variations were high. Therefore, it was the demand that had the biggest impact on the model.

More sampling would be needed to establish an accurate assessment, but in general, it appears from our model that allowing a schedule delay to reduce rework associated with design changes results in nearly double the savings provided by allowing extra tooling revisions.