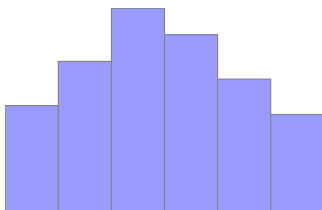


A Branch-and-Price Approach to Product Clustering & Machine Setup in Printed Circuit Board Assembly

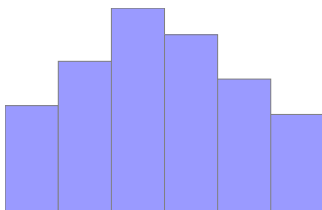
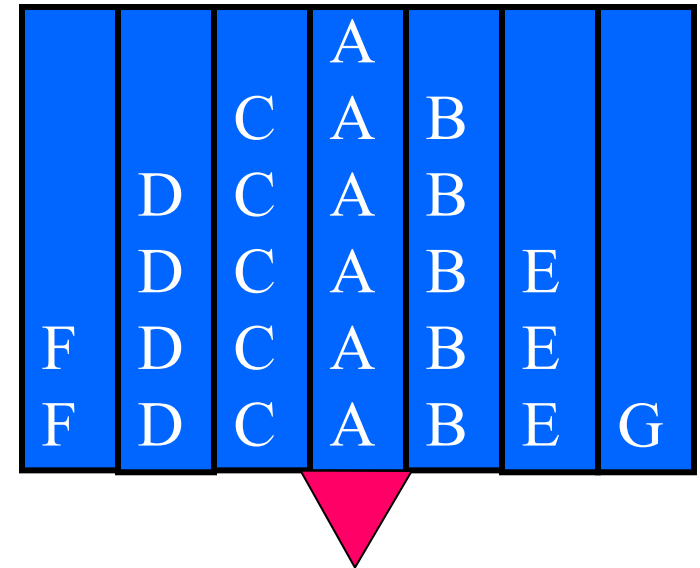
INFORMS -- Salt Lake City, 2000

Amy Mainville Cohn, MIT
Michael J. Magazine, Univ. of Cincinnati
George G. Polak, Wright State Univ.



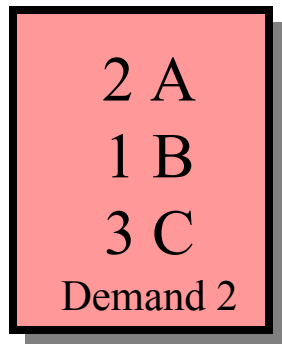
The PCB Problem

- Components to be placed on a PCB are retrieved from a pick-and-place machine
- Board is automatically positioned under insertion head
- Components are retrieved from their sleeves and placed by the insertion head
- Processing time is linear in distance traveled from insertion point
- Total processing time depends on which component is placed in which sleeve

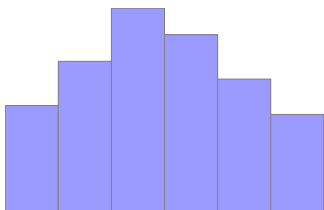
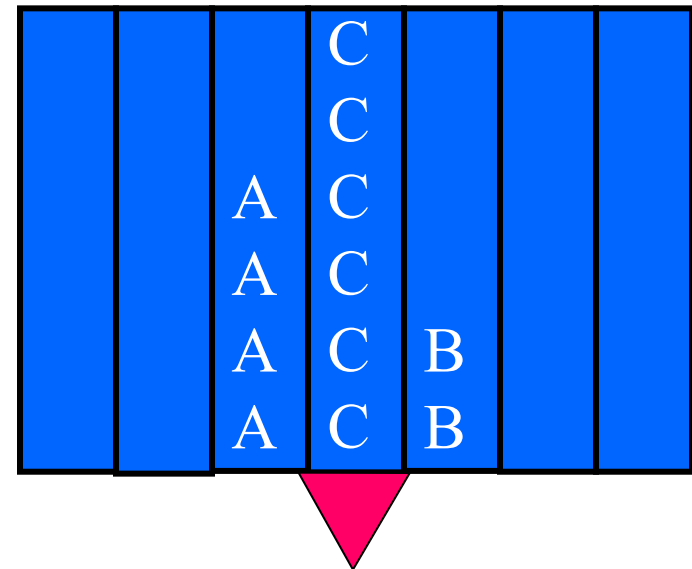


The PCB Problem

Given one board type, problem is trivial

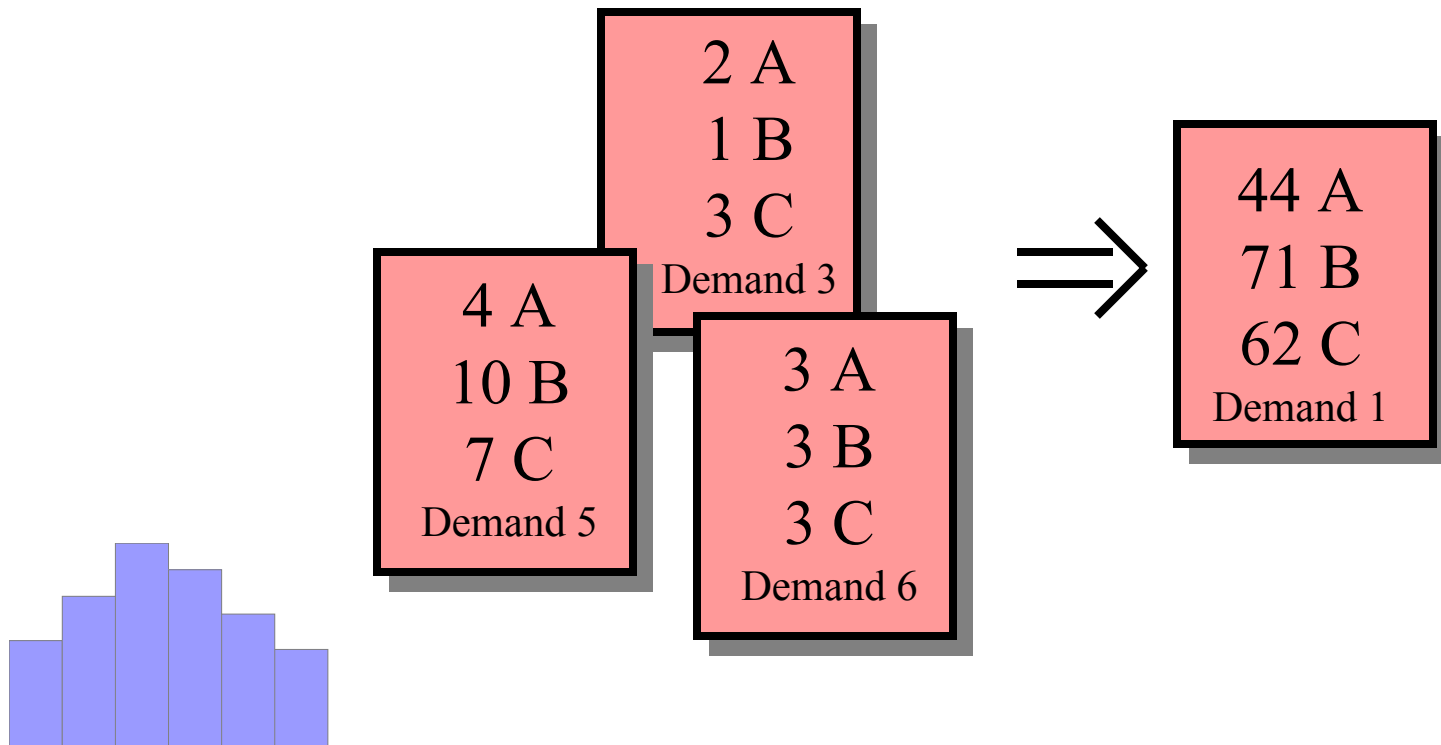


=>



The PCB Problem

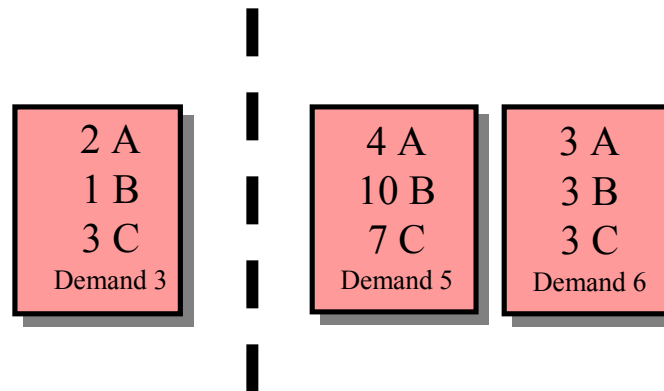
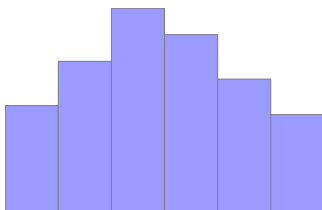
- Given multiple board types, no setup cost ($\sigma = 0$), decomposes by board type
- Given multiple board types, high setup cost ($\sigma = \infty$), group all components and rank



The PCB Problem

Problem Statement:

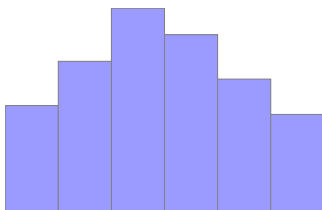
Given a collection of K board types comprised of P component types, loaded into P machine sleeves with setup cost σ , partition the board types into clusters and for each cluster configure the pick-and-place machine so as to minimize overall cost (setup + processing)



PCB Planning Literature

Product Clustering, Machine Setup,
Component Placement & other problems
generally studied as separate units

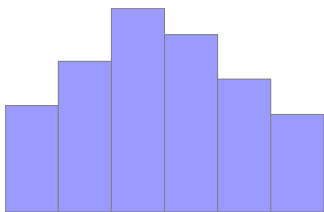
- Bard, Clayton & Feo (1994), Machine Setup and Component Placement in Printed Circuit Board Assembly, *Int Jn. Flex. Mfg*, 6
- Ben-Arieh, and Chang (1994), An Extension to the p-Median Group Technology Problem, *Comp & Ops Res* 21
- Crama, van de Klundert, and Spieksma (1999), Production Planning Problems in PCB Assembly, *GEMME* 9925
- Hashiba and Chang (1991), PCB Assembly Setup Using Group Technology, *Proc. 13-th Ann. Conf. Computers & Ind. Eng.*
- Hillier, M.S. and M.L. Brandeau (1998), Optimal Component Assignment and Board Grouping in PCB Manufacturing, *Ops Res* 46.
- Jain, Johnson & Safai (1996), Implementing setup optimization on the shop floor, *Ops. Res.* 43



Integrating Product Clustering and Machine Setup

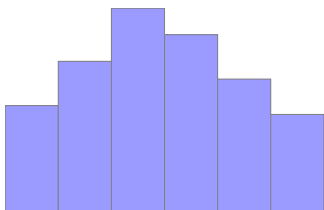
Magazine and Polak formulated an **integrated** product clustering and machine setup model

- IP is computationally challenging -- largest instance solved to optimality has 11 board types, 16 component types
- Model is lightly constrained, so not well suited to Constraint Logic Programming



New Integrated Approach

- Our strategy is to formulate the clustering aspect as a set partitioning problem
- The cost of a cluster is its optimal configuration cost plus setup
- Able to solve problems with more board types and more components in less time than the previous integrated approach



Notation

S is the set of board types

$$|S| = K$$

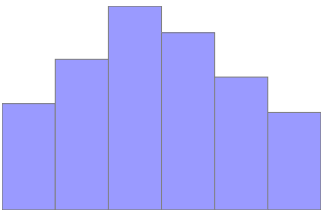
Total number of potential clusters is $2^K - 1$

$f_c^*(\bullet)$ is the optimal processing cost function associated with cluster c

$f_c^*(d)$ is the processing cost for set d of using the optimal configuration of cluster c

$$f_c^*(c) = \sum_{b_i \in c} f_c^*(b_i)$$

$$f_{\bullet}^*(b_i) \geq f_{b_i}^*(b_i)$$



Set Partitioning Formulation

- Each cluster c is represented by a column
- Choose a set of columns such that each board type is in exactly one column

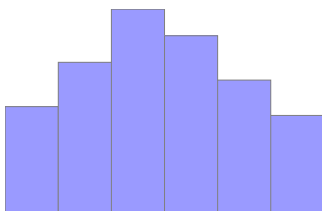
- Formulation:

$$\text{Min} \sum [\sigma + f_c^*(c)] x_c$$

st

$$\sum_{c: b_i \in c} x_c = 1 \quad \text{for any board type } b_i$$

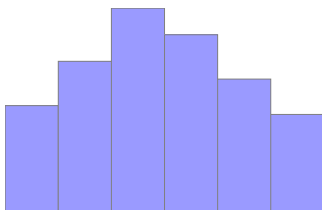
$$x_c \in \{0,1\} \quad \text{for any cluster } c$$



Board 1:	1	0	0	0	1	1	1	0	0	0	1	1	1	0	1	= 1 (π_1)
Board 2:	0	1	0	0	1	0	0	1	1	0	1	1	0	1	1	= 1 (π_2)
Board 3:	0	0	1	0	0	1	0	1	0	1	1	0	1	1	1	= 1 (π_3)
Board 4:	0	0	0	1	0	0	1	0	1	1	0	1	1	1	1	= 1 (π_4)

Set Partitioning Formulation

- Problem: IP with 2^K columns!
- Solve using **branch-and-price** -- branch-and-bound with dynamic column generation at each node
- Column generation solves a pricing problem to identify negative reduced cost column
- Reduced cost of a column is $\sigma + f_c^*(c) - \sum_{b_i \in c} \pi_i$, where π_i is the dual associated with the cover constraint for the i^{th} board type -- this can also be written as $\sigma + \sum_{b_i \in c} (f_c^*(b_i) - \pi_i)$



Set Partitioning Formulation

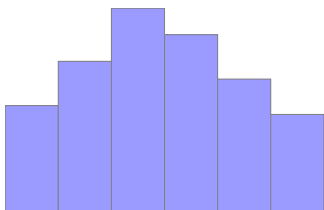
- Pricing problem: find the cluster c^* that minimizes

$$p^* \equiv \sigma + \sum_{b_i \in c} (f_c^*(b_i) - \pi_i)$$

st

$$c \subseteq S \equiv \{b_1, b_2, \dots, b_K\}$$

- If $p^* < 0$, then we have identified a new pivot column
- If $p^* \geq 0$, then we are optimal for the LP relaxation
- How do we solve this sub-problem?

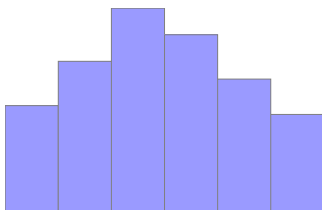


Solving the Subproblem by IP

- One approach is to formulate as an IP
- Multiple formulations exist
- Enabled us to solve some problems with up to 24 boards
- Still too slow

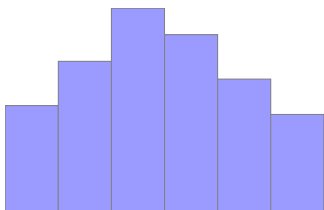
Problems:

- Since all clusters are feasible, difficult to prune IP's
- x_{cs} -- matching variables -- component c in sleeve s -- binary variable that grows with the square of the number of component types -- fractional solutions



IP Approach I -- Demand Model

- **Selection variables:** $y^k = 1$ if board type k included, else 0
- **Assignment variables:** $x_{ij} = 1$ if component i in sleeve j , else 0
- **Demand variables:** $t_i^k =$ number of components i to be processed for board type k -- will be 0 or the total demand for i by k
- **Processing variables:** $z_{ij} =$ number of components i processed in sleeve j
- **Objective coefficients:** p_j is the cost of processing one unit from sleeve j



IP Approach I -- Demand Model

$$\text{Minimize} \quad z = \sigma + \sum_{j \in N} \sum_{i \in N} p_j z_{ij} - \sum_{k \in K} \pi^k y^k$$

s.t.

$$\sum_{i \in N} x_{ij} = 1, \forall j \in N,$$

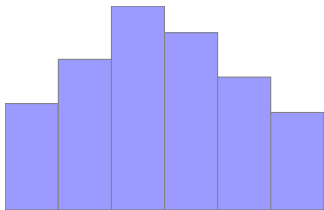
$$\sum_{j \in N} x_{ij} = 1, \forall i \in N,$$

$$t_i^k = d_i^k y^k, \forall i \in N, k \in K,$$

$$\sum_{j \in N} z_{ij} = \sum_{k \in K} t_i^k, \forall i \in N,$$

$$z_{ij} \leq M x_{ij}, \forall i, j \in N,$$

$$x_{ij}, y^k \in \{0,1\}, t_i^k, z_{ij} \geq 0, \forall i \in N, j \in N, k \in K.$$



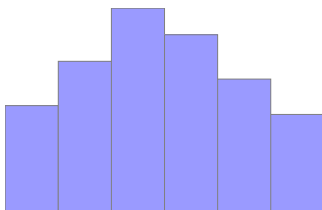
IP Approach I -- Demand Model

Pros:

- Pricing problem as an IP
- Clear formulation

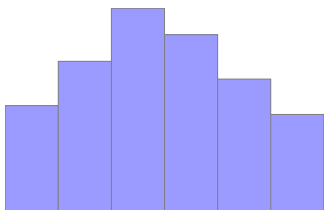
Cons:

- Large number of binary variables
- Large number of constraints
- Frequent fractional solutions



IP Approach II -- W Model

- **Selection variables:** $y^k = 1$ if board type k included, else 0
- **Assignment variables:** $x_{ij} = 1$ if component i in sleeve j , else 0
- **Processing variables:** $w_{ij}^k = 1$ if component i processed from sleeve j for board type k
- **Objective coefficients:** δ_{ij}^k is the cost of processing all components of type i from sleeve j for board type k



IP Approach II -- W Model

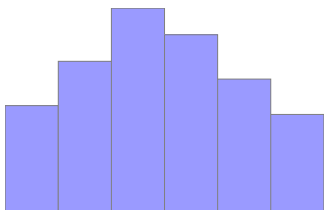
$$\text{Minimize} \quad z = \sigma + \sum_{k \in K} \sum_{j \in N} \sum_{i \in N} \delta_{ij}^k w_{ij}^k - \sum_k \pi^k y^k$$

s.t.

$$\sum_{i \in N} x_{ij} = 1, \forall j \in N, \quad \sum_{j \in N} x_{ij} = 1, \forall i \in N,$$

$$\sum_{i \in N} w_{ij}^k = y^k, \forall j \in N, k \in K, \quad \sum_{j \in N} w_{ij}^k = y^k, \forall i \in N, k \in K,$$

$$w_{ij}^k \leq x_{ij}, \forall i, j \in N, k \in K, \quad x_{ij}, w_{ij}^k, y^k \in \{0, 1\}, \forall i \in N, j \in N, k \in K.$$



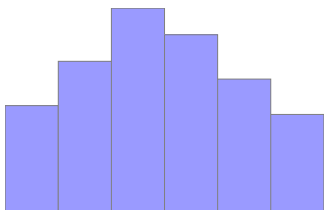
IP Approach II -- W Model

Pros:

- For a given vector y , the x 's and w 's can be relaxed
- Fewer binary variables than approach 1

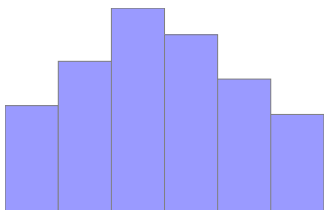
Cons:

- All y vectors are feasible
- No pruning due to infeasibility
- Can prove that the LP relaxation at any node will be fractional
- Need to traverse full depth of the tree for an integer solution



IP Approach III -- Big M Model

- **Selection variables:** $y^k = 1$ if board type k included, else 0
- **Assignment variables:** $x_{ij} = 1$ if component i in sleeve j , else 0
- **Contribution variables:** θ^k is the contribution from board type k to the reduced cost objective function
- **Cost coefficient:** δ_{ij}^k is the cost of processing components of type i from sleeve j for board type k



IP Approach III -- Big M Model

Maximize ε

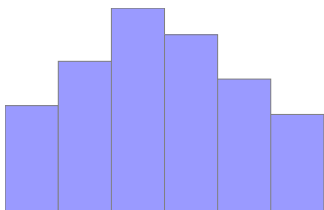
s.t.

$$\sum_{i \in N} x_{ij} = 1, \forall j \in N, \quad \sum_{j \in N} x_{ij} = 1, \forall i \in N,$$

$$-\pi^k + \sum_{j \in N} \sum_{i \in N} \delta_{ij}^k x_{ij} \leq M(1 - y^k) - \theta^k, \forall k \in \{1, \dots, K\}.$$

$$\theta^k \leq M y^k, \forall k \in \{1, \dots, K\}, \quad \sum_{k \in K} \theta^k \geq \sigma + \varepsilon,$$

$$\varepsilon \geq 0; \quad x_{ij}, y^k, \theta^k \geq 0, \quad \in \{0, 1\}, \forall i \in N, j \in N, k \in K.$$



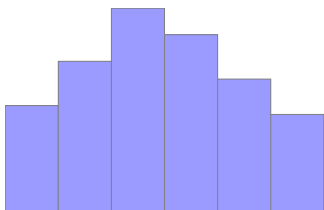
IP Approach III -- Big M Model

Pros:

- Can solve many problems that couldn't be solved before
- More pruning by infeasibility

Cons:

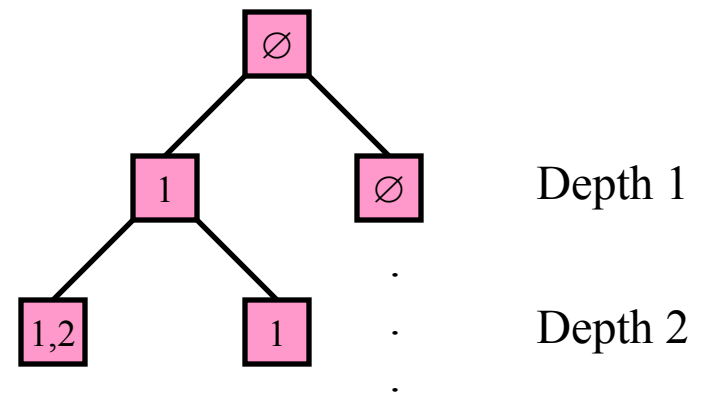
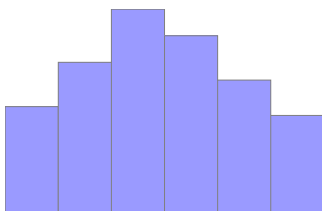
- Fractional solutions -- even with tightening of Big M
- All approaches have matching component -- at least N^2 binary variables
- Don't take advantage of simple cost computation (sorting)
- Still too slow!



Combinatorial Approach

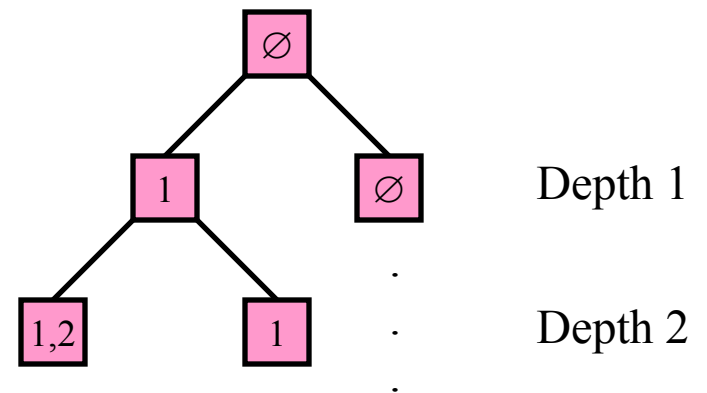
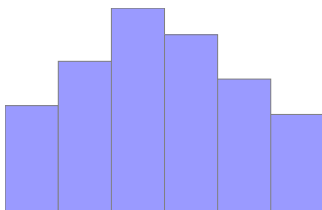
Enumeration with bounding:

- Rank board types: 1, 2, 3, 4, ...
- Each depth i of the tree is associated with the decision to include or exclude board b_i from the new cluster
- Value of a node is the reduced cost of the cluster it represents
- Can prune at a node if we know that none of its offspring can have negatived reduced cost



Combinatorial Approach

- Takes advantage of simple cost function
- Can generate many columns per iteration
- Exponential number of clusters -- need strong bounding



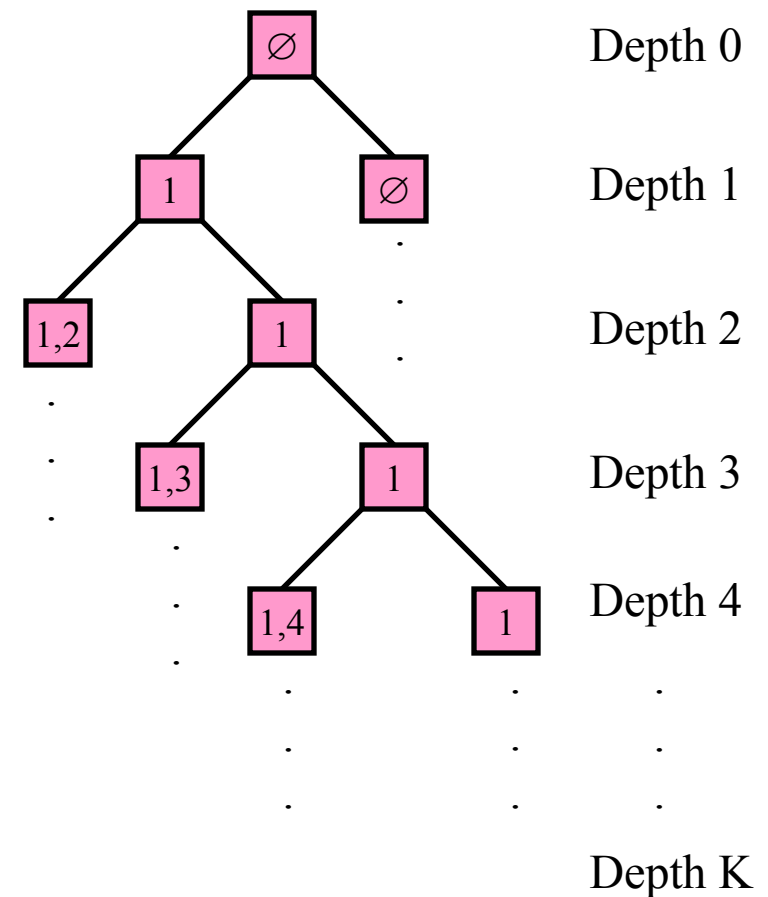
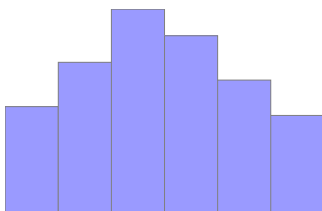
Combinatorial Approach

- Reduced cost of node $\{1,3\}$ is

$$r_c(\{1,3\}) \equiv \sigma + f_{1,3}^*(1) + f_{1,3}^*(3) - \pi_1 - \pi_3$$
- By adding further board types to this set, at most we can decrease by

$$\sum_{\substack{\text{Depth}(3) < \\ \text{Depth}(i) \leq K}} \pi_i$$

- If $r_c(\{1,3\}) - \sum_{\substack{\text{depth}(3) < \\ \text{depth}(i) \leq K}} \pi_i \geq 0$,
 we can prune from the node



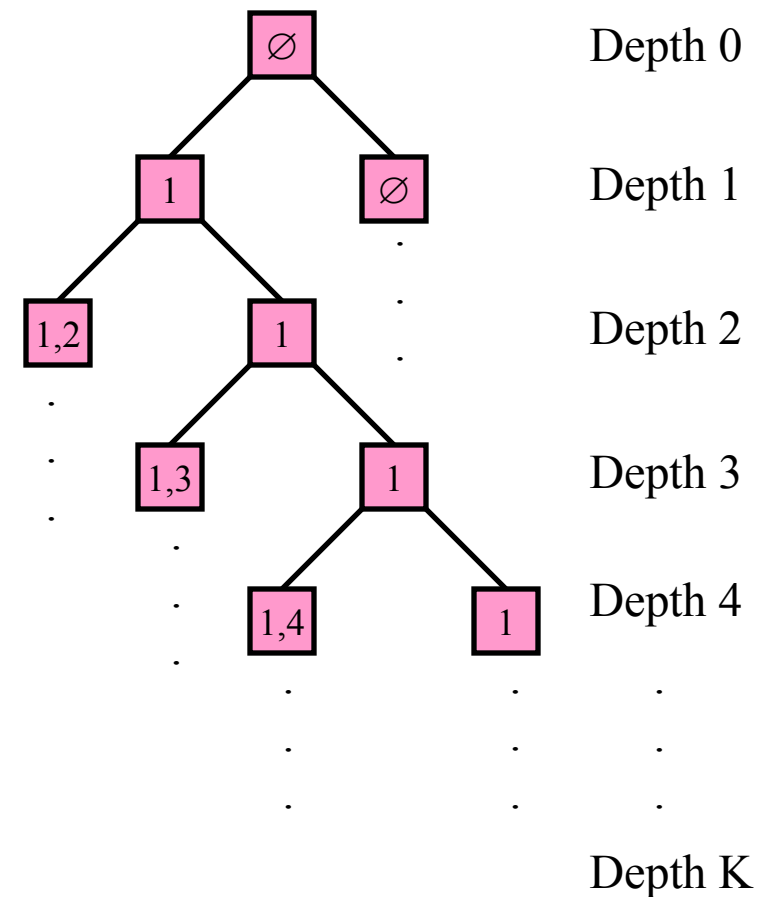
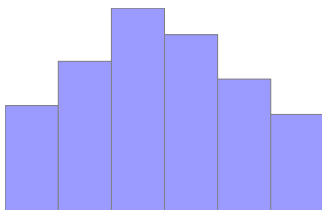
Combinatorial Approach

- Can we tighten this?
- Yes -- each board type b_i added to the current cluster will decrease the reduced cost by its dual but will also increase the reduced cost by its processing cost -- this will be at least $f_{bi}^*(b_i)$

- If $r_c(\{1,3\}) + \sum_{\substack{\text{depth}(3) < \\ \text{depth}(i) \leq K}} \min[0, f_i^*(i) - \pi_i] \geq 0$

we can prune the tree

- Solves many problems up to 32 board types



Combinatorial Approach

Bound can still be tightened. Example:

Let V be the reduced cost of the board types included in the node representing set N and assume two board types remain to be considered -- J, K .

What is the most we can decrease V by?

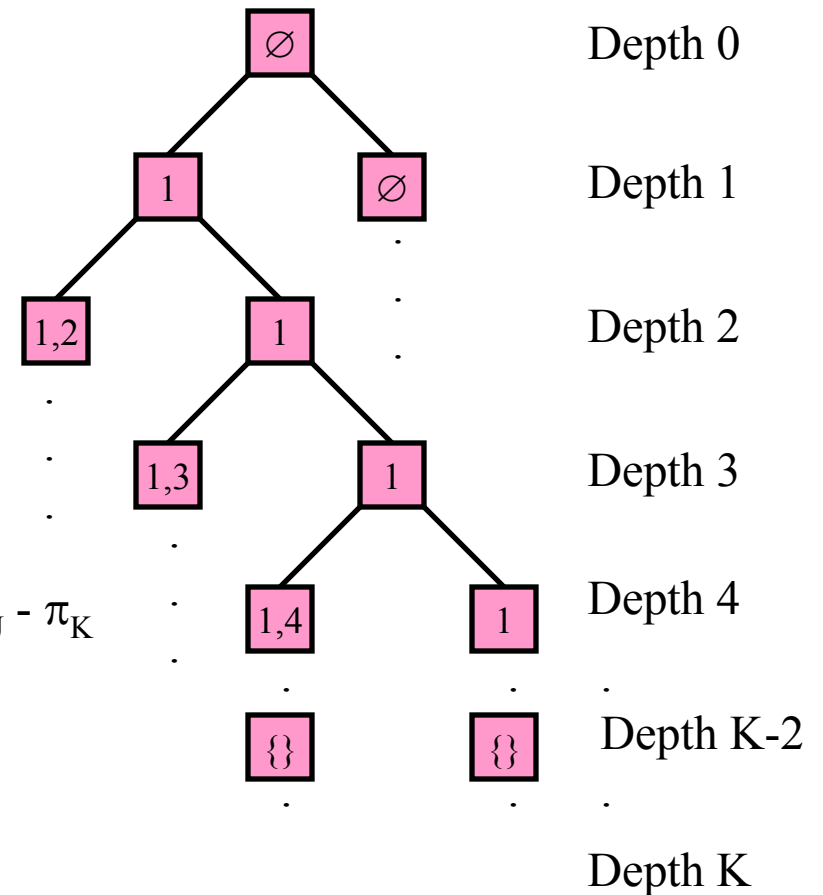
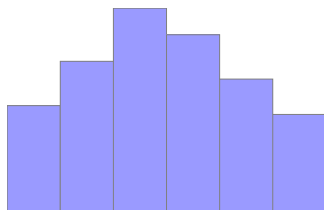
Add neither: lower bound is V

Add J to N : lower bound is $V + f_J^*(J) - \pi_J$

Add K to N : lower bound is $V + f_K^*(K) - \pi_K$

Add both: lower bound is $V + f_{\{J,K\}}^*(J) + f_{\{J,K\}}^*(K) - \pi_J - \pi_K$

Take minimum of these four



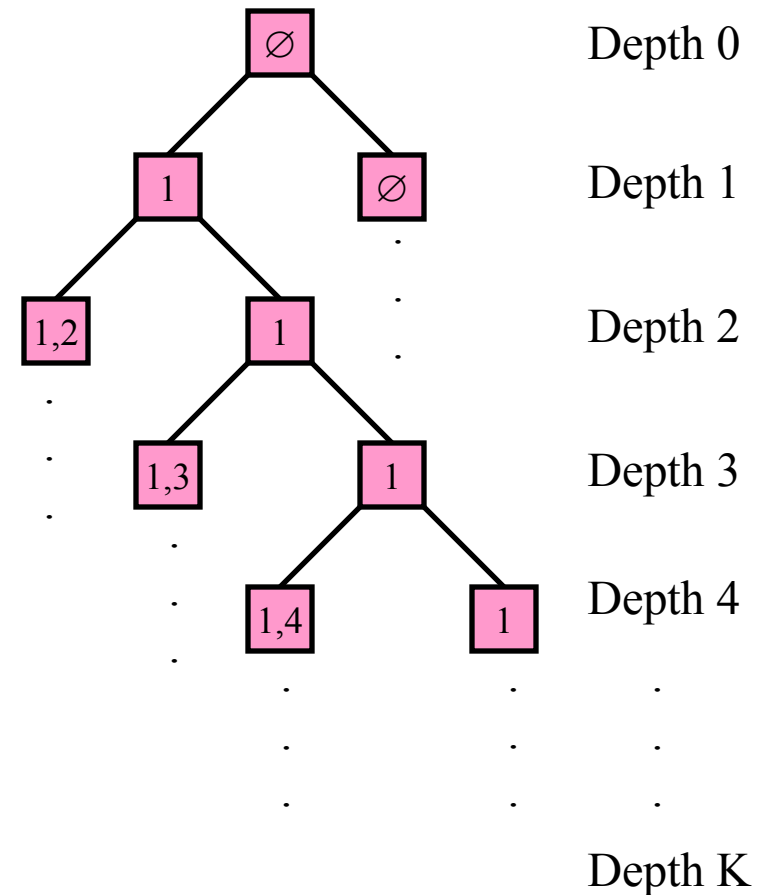
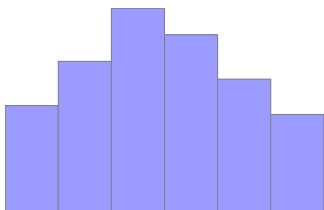
Combinatorial Approach

How can we generalize this?

For any node at depth i , the remaining potential (i.e. most negative possible contribution) from the remaining board types (depth $i+1$ to k) is found by solving

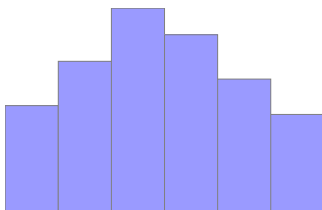
$$\begin{aligned} \text{Min} \sum_{b_i \in c} f_c^*(b_i) - \pi_i \\ \text{st } c \subseteq \{b_{i+1}, b_{i+2}, \dots, b_k\} \end{aligned}$$

This is simply the pricing problem on a reduced set of board types with setup cost 0!



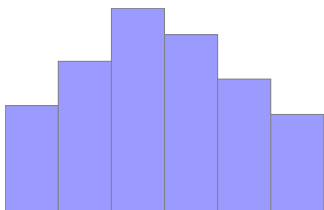
Computational Results

- Considered 115 instances
- Instances comprised of 10 to 40 board types and 16 to 100 components; range of setup costs for each instance
- 87 problems solved to **integer** optimal solutions at the root node (19 more solved to fractional solutions)



Time to Integer Optimality

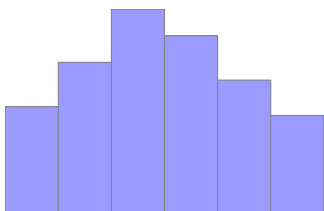
Boards x Components	Time Range (seconds)
10 x 16, 10 x 100	≤ 6
16 x 100	5 - 365
24 x 16	1 - 2,550
32 x 16	2 - 121,583
32 x 100	18 - 132,033
40 x 16	6 - 22,948



* HP9000 Model D370 running HPUX 10.20 and CPLEX 6.0

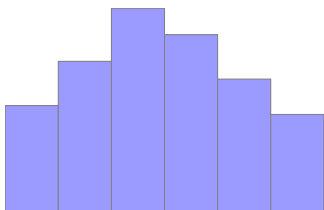
Computational Results

- Can find integer solutions to instances of up to 100 board types easily if setup cost is small enough (e.g. optimal solution has clusters of 3 or 4 boards)
- Most slow instances found integer optimal solution early, then many degenerate pivots
- Many unsolved instances appeared to be cycling



Further Research

- Tighter bounds may still be found
- Perturbation might help with degeneracy
- Better column management could lead to solving some of the unsolved problems



Conclusions

- Enables us to simultaneously cluster objects and “process” clusters to find exact solutions
- Combinatorial approach allows us to exploit cost function
- Approach is extendable to many other manufacturing problems of a similar nature
- Wide range of existing theory for set partitioning, column generation, branch-and-price, as well as specifics of each application’s cost function

