

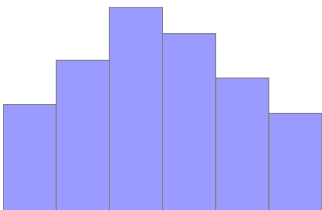
Printed Circuit Board Manufacturing and Large-Scale Optimization Techniques

M&SOM Conference – USC, June 2003

Amy Mainville Cohn, Sarah Root, David Stajninger, and
Reid Tatoris (U of Michigan)

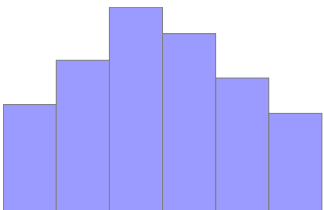
Michael Magazine (U of Cincinnati)

George Polak (Wright State U)



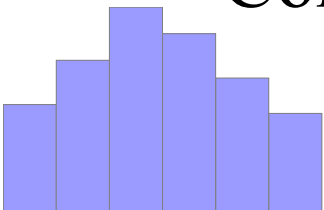
Goals

- Present a technique for solving a large integrated PCB planning problem
- Discuss how ideas of merging LP techniques with enumerative algorithms can be generalized to other applications in manufacturing and service operations

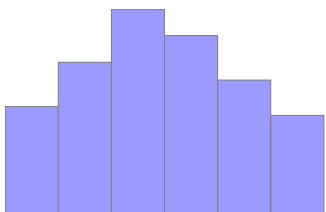
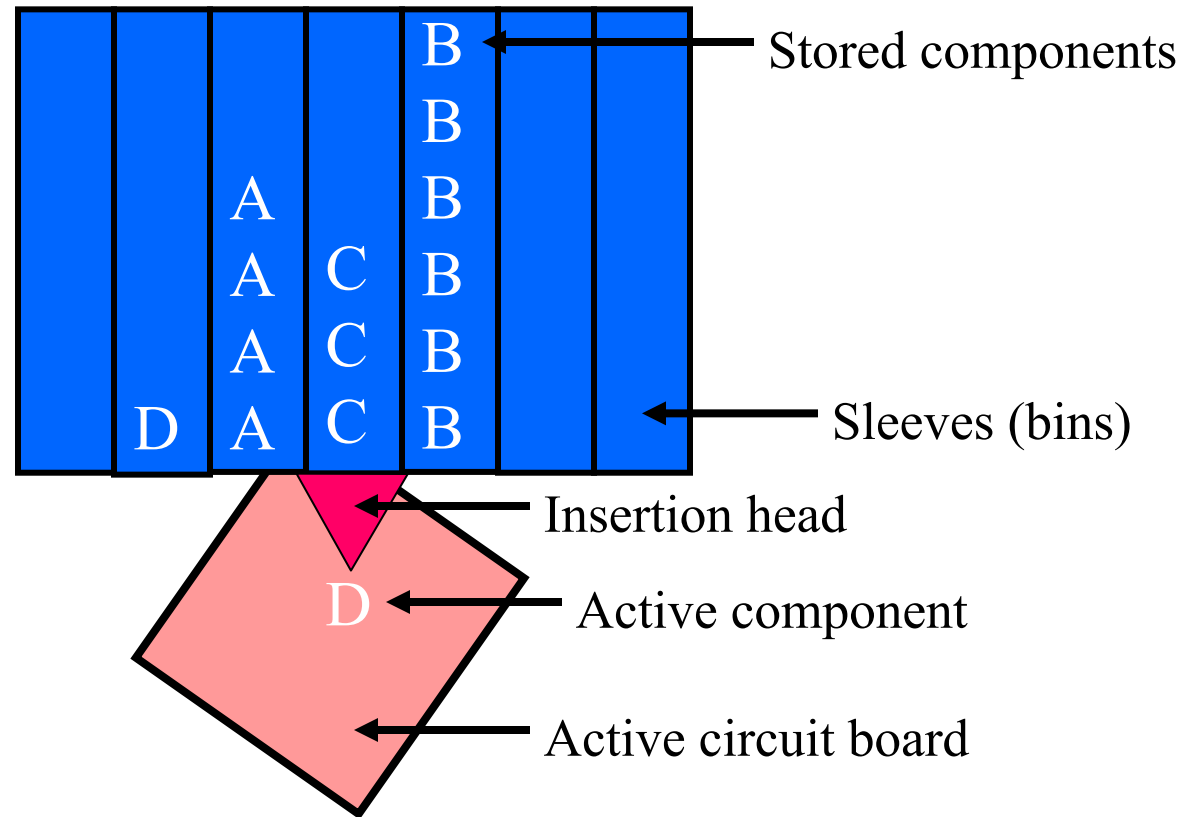


Outline

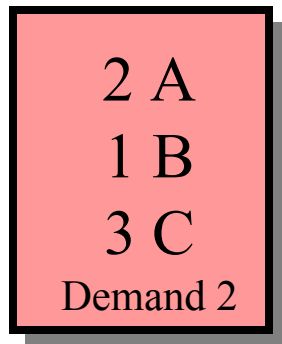
- Introduce *integrated product clustering and machine setup problem*
- Special cases
- Set partitioning formulation
- Branch-and-price approach
- Specialized algorithm for pricing problem
- Computational results
- Conclusions and future research



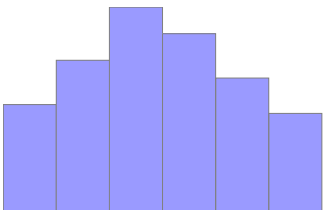
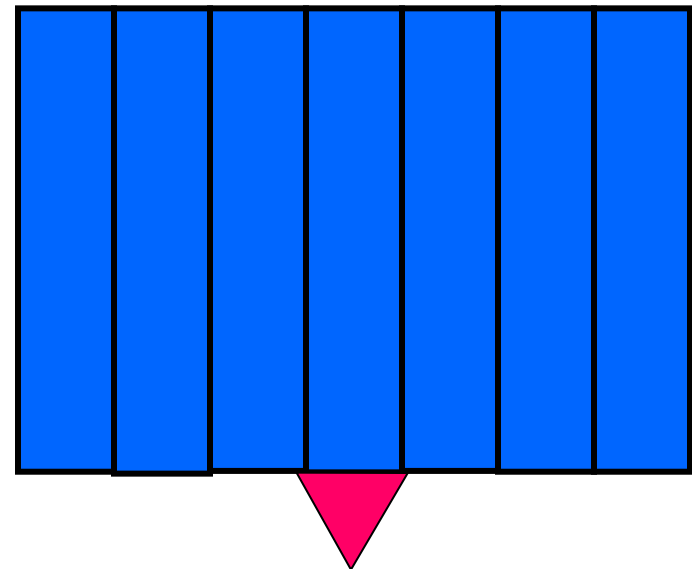
Pick-and-Place Technology



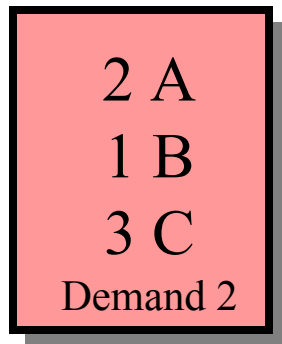
Special Cases: 1 Board Type



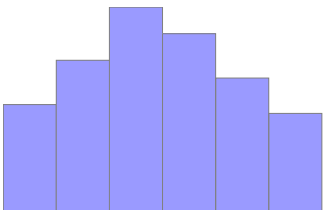
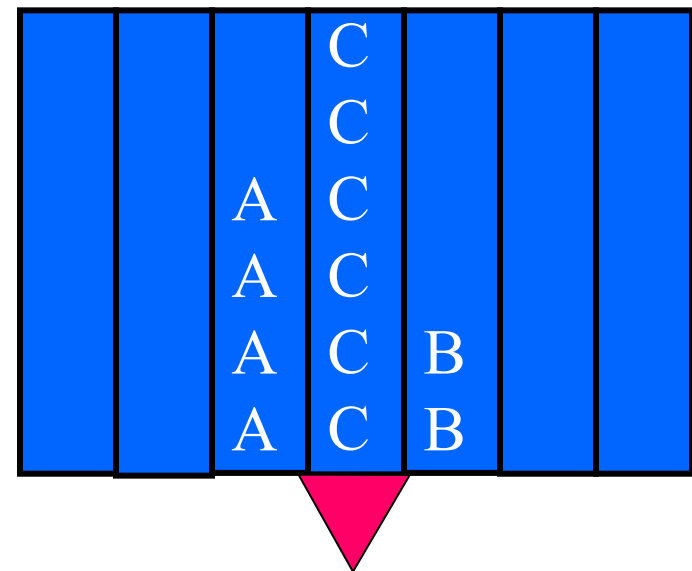
=>



Special Cases: 1 Board Type

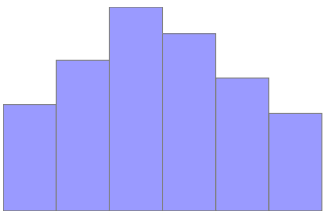


=>



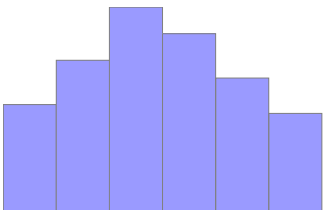
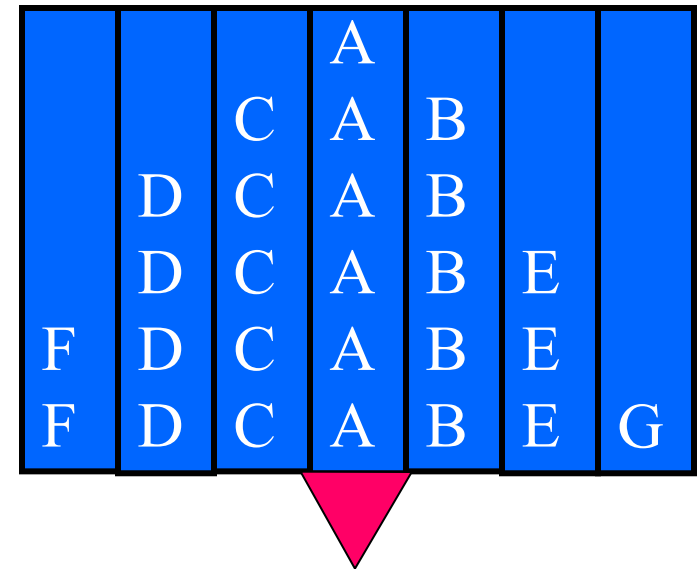
Special Cases: 1 Board Type

Given one board type,
problem is trivial – rank
the components and put
the most popular
components in the least
costly sleeves – “pipe-
organ configuration”
(Hardy, Littlewood, and
Polya, 1952)



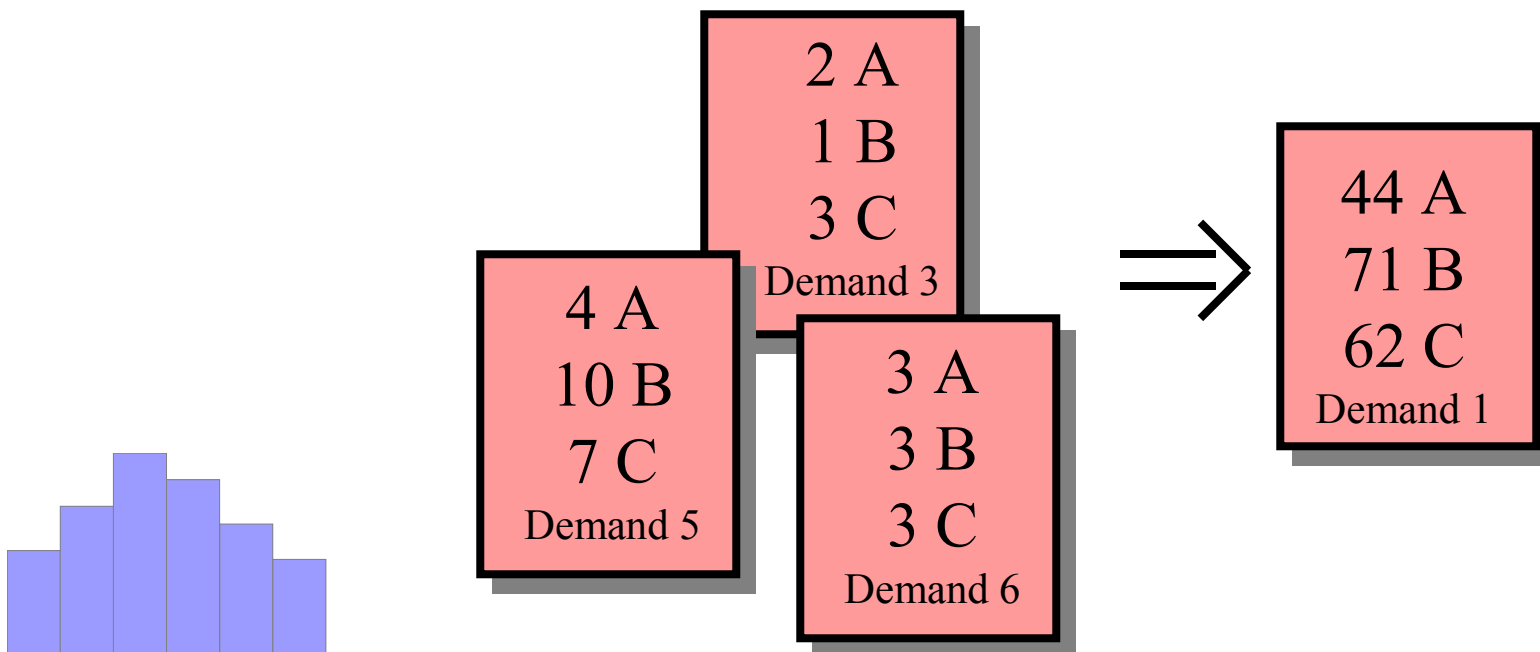
Special Cases: 1 Board Type

Given one board type,
problem is trivial – rank
the components and put
the most popular
components in the least
costly sleeves – “pipe-
organ configuration”
(Hardy, Littlewood, and
Polya, 1952)



Special Cases: Extreme Values of σ

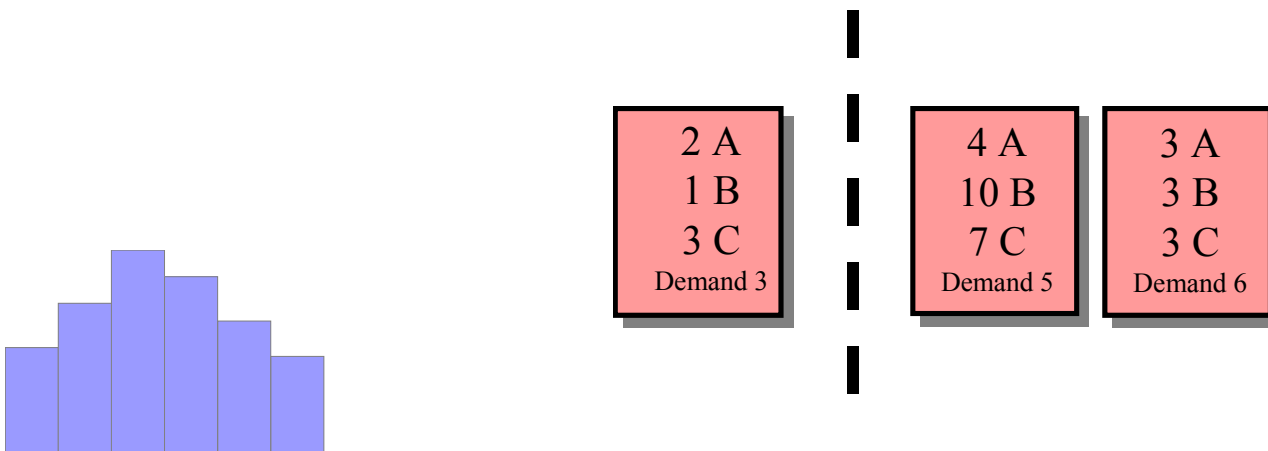
- Given multiple board types, no setup cost ($\sigma = 0$), decomposes by board type
- Given multiple board types, high setup cost ($\sigma = \infty$), group all components and rank



General Case: Moderate Values of σ

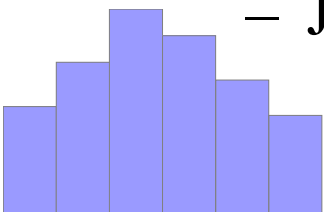
Problem Statement:

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



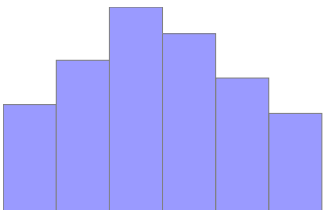
Literature Review

- Product clustering, machine setup, component placement, and other problems have generally been studied as separate units or with limited integration
 - Bard, Clayton & Feo (1994)
 - Ben-Arieh and Chang (1994)
 - Crama, van de Klundert, and Spieksma (1999)
 - Hashiba and Chang (1991)
 - Hillier, M.S. and M.L. Brandeau (1998)
 - Jain, Johnson & Safai (1996)



Literature Review, cont.

- Magazine and Polak formulated an integrated product clustering and machine setup model in 2002
- IP is computationally challenging – largest instance solved to provable optimality had 11 board types, 16 component types



Set Partitioning Formulation

- Each cluster c is represented by a variable (i.e. column) x_c
- $f_c^*(c)$ is the optimal processing cost for cluster c
- Choose a set of columns such that each board type is in exactly one column

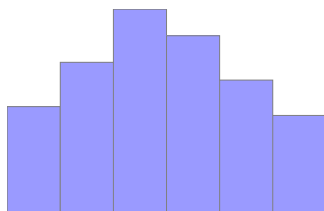
Formulation:

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

st

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

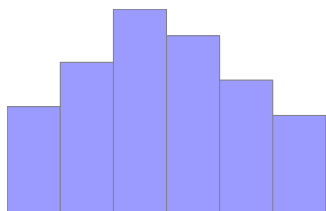
$$x_c \in \{0,1\} \quad \text{for any cluster } c$$



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)

Branch-and-Price Approach

- Problem: IP with 2^B 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



Pricing Problem

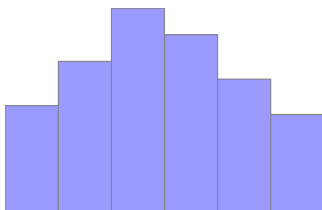
- Reduced cost of column c is $\sigma + f_c^*(c) - \sum_{b \in c} \pi_b$, where π_b is the dual associated with the cover constraint for the b^{th} board type – this can also be written as $\sigma + \sum_{b \in c} (f_c^*(b) - \pi_b)$
- Pricing problem: find the cluster c^* that minimizes

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

st

$$c \subseteq \{b_1, b_2, \dots, b_B\}$$

- 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?



Sub-Problem Formulation 1

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

s.t.

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

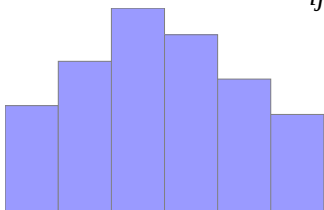
$$\sum_{j \in N} x_{ij} = 1, \forall i \in N,$$

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

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

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

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



Sub-Problem Formulation 2

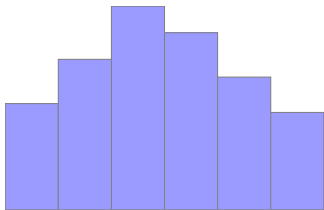
$$\text{Minimize} \quad z = \sigma + \sum_{k \in K} \sum_{j \in N} \sum_{i \in N} \delta_{ij}^k w_{ij}^k - \sum_k \pi^k y^k$$

s.t.

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

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

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



Sub-Problem Formulation 3

Maximize ε

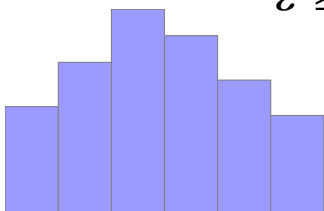
s.t.

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

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

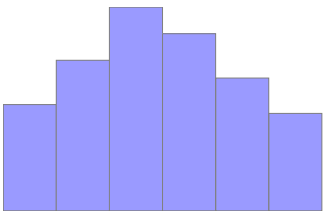
$$\theta^k \leq M y^k, \forall k \in \{1, \dots, K\}, \quad \sum_{k \in K} \theta^k \geq \sigma + \varepsilon,$$

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



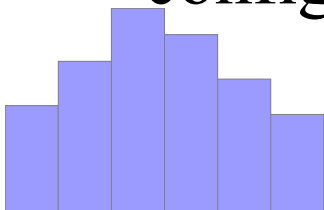
Source of Computational Difficulty

- In all cases, our sub-problem is itself an IP with a large number of binary variables
- Furthermore, the LP relaxation is very weak
 - requires substantial branching
- Why?
 - To compute the cost



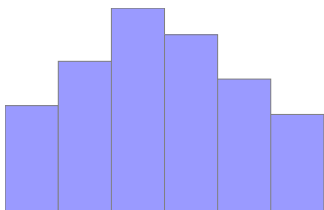
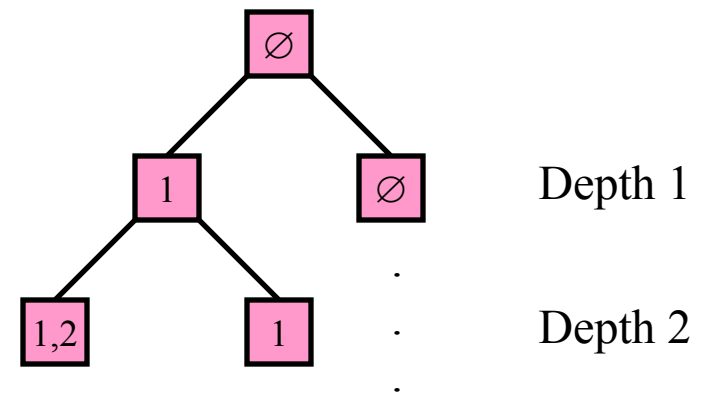
Combinatorial Approach: Motivation

- In the IP approaches to the pricing problem, we make two types of decisions
 - Which boards to include
 - How to configure the pick-and-place machine
- Weak LP relaxations tend towards enumerating *all* clusters, then finding their optimal configuration
- But for a given cluster, we know the optimal configuration!



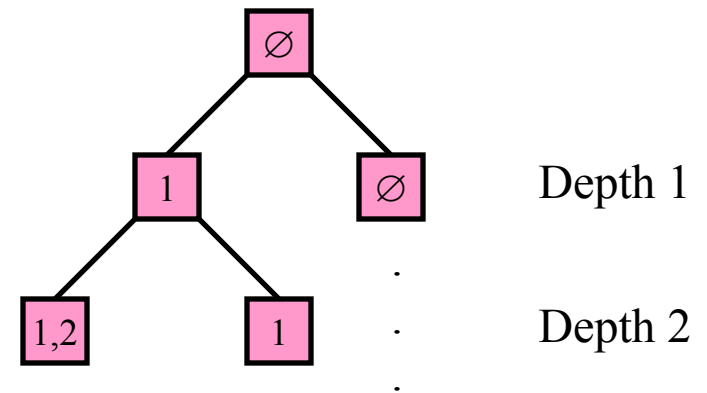
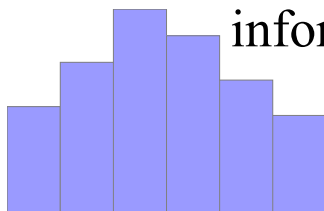
Rank-Cluster-and-Prune

- Rank board types
- Each depth b of the tree is associated with deciding whether to include board b in the new cluster
- The value of a node is the reduced cost of the cluster it represents



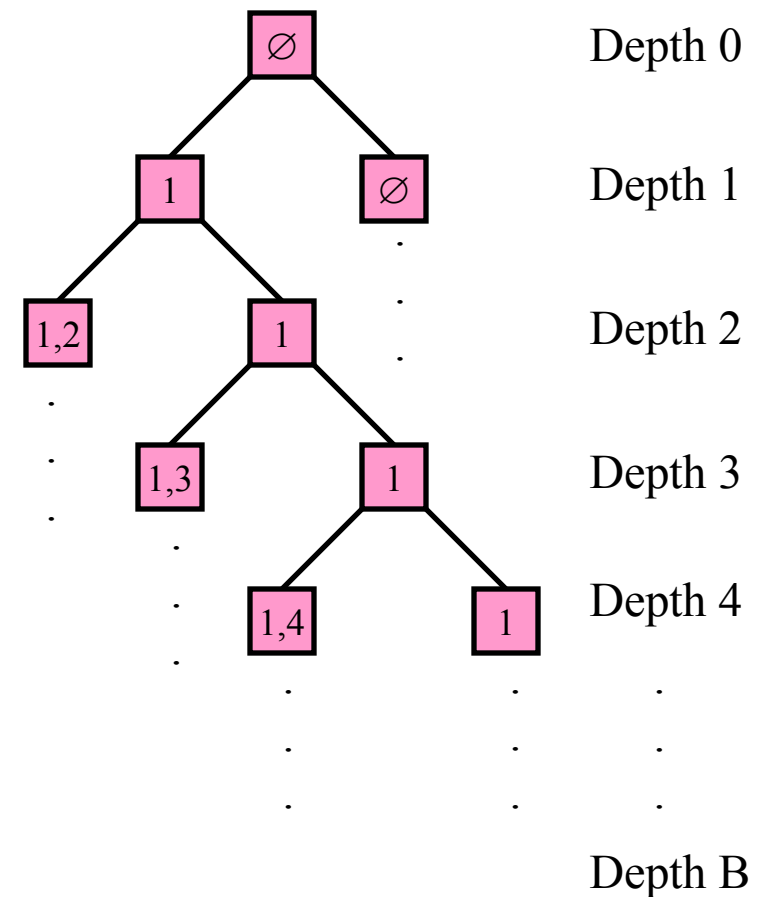
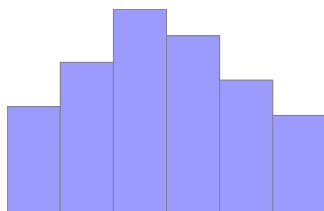
Trade-Offs of Combinatorial Approach

- Pros:
 - Takes advantage of simple cost function
 - Can generate many columns per iteration
- Cons:
 - Exponential number of clusters – need strong bounding
 - How can we leverage dual information?



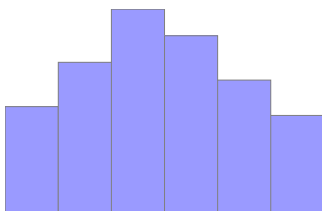
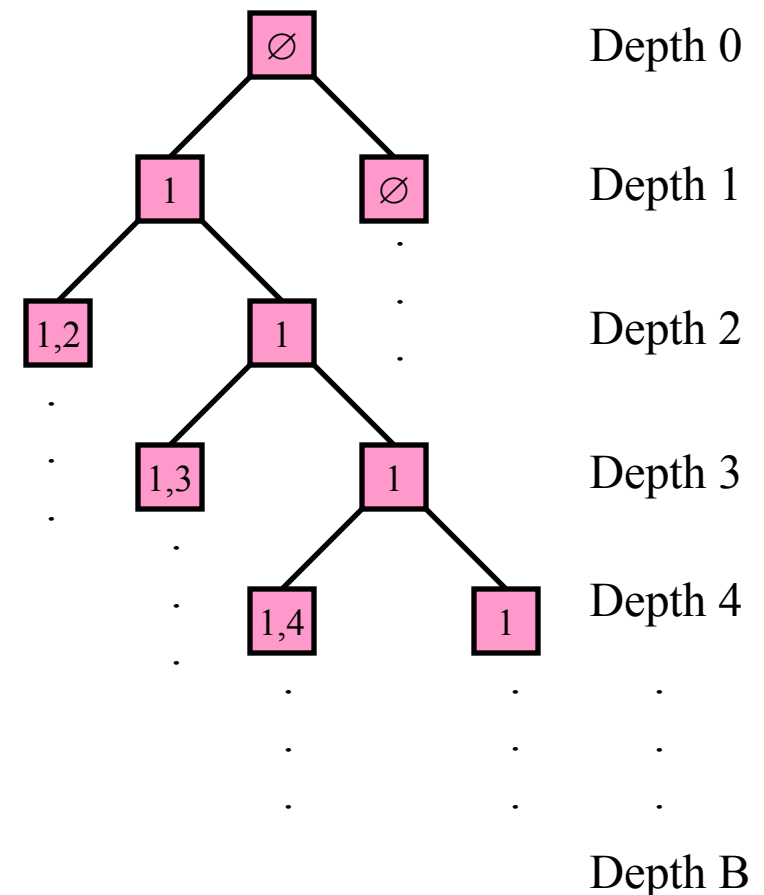
Pruning Techniques

- Adding a board to the cluster increases reduced cost by its processing cost but (potentially) decreases by its dual
- If $f_b^*(b) \geq \pi_b$, we can ignore board type b



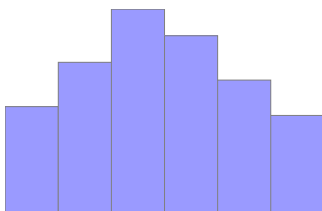
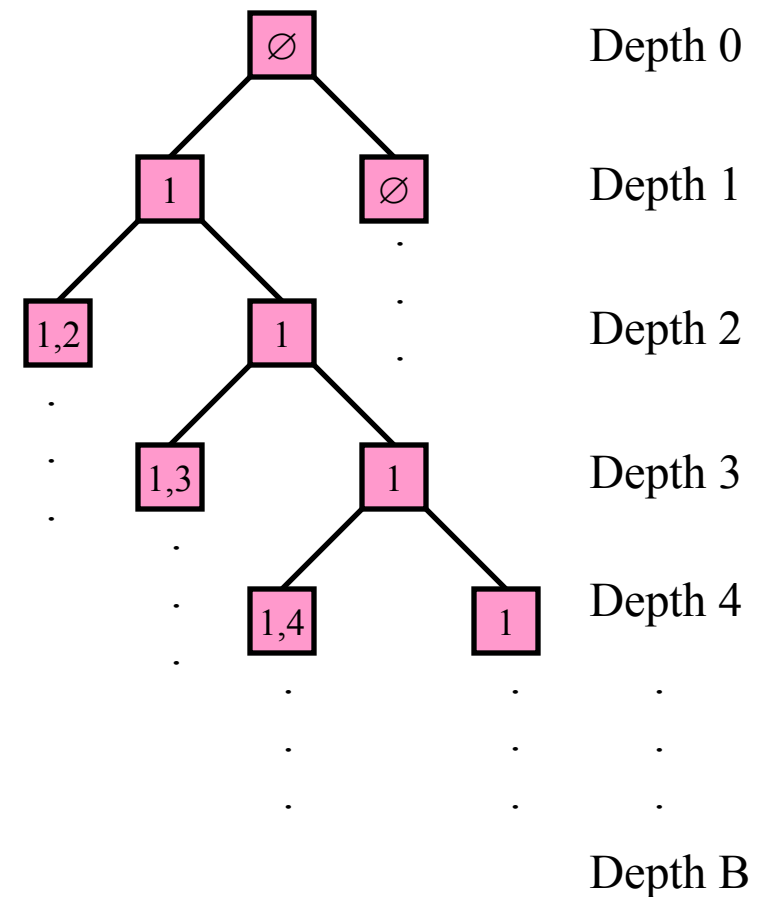
Pruning Techniques

- If the current node at depth d has positive value V , the duals of the remaining boards contributed (less their manufacturing cost) must outweigh the current cost
- If $\sum_{b>d} (f_b^*(b) - \pi_b) \geq -V$, we can never find a negative reduced cost cluster in the current branch and thus can prune



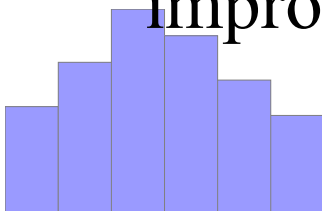
Pruning Techniques

- Additional, more sophisticated pruning techniques exist based on solving the pricing problem to optimality over a subset of the remaining board types
- Enabled use to greatly expand the size of tractable problem instances



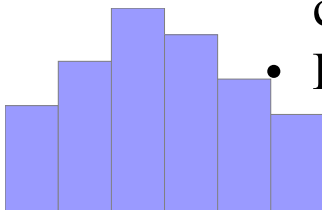
Computational Results

- Prior approach solved problems of at most 11 boards and 16 components
- Set partitioning formulation with IP sub-problem was intractable for all instances
- Our approach, in comparable time, solves problems ranging up to 100 boards, 100 components
- Still plenty of need (and opportunity?) for improvement!



Conclusions

- SP formulation, b-and-p algorithm, RCP pricing approach allow us to solve many new problem instances
- More generally:
 - Set partitioning allows us to simplify the model
 - B+P allows us to make large models tractable, leverage dual information
 - Combinatorial pricing approach may be a better way to generate columns
 - In our case, lets us leverage easy-to-compute (but non-linear) cost function
 - Pruning opportunities are critical!



Future Research

- Generalized code for b-and-p implementation of set partitioning problems
- Addressing degeneracy
- Tightening bounds in RCP
- Expanding problem scope

