Unit Load Storage Systems Design

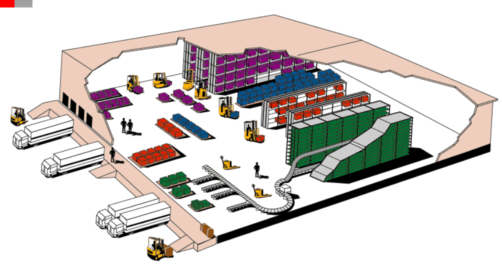
Marc Goetschalckx

Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0205
(404) 894-2317
marc.goetschalckx@isye.gatech.edu

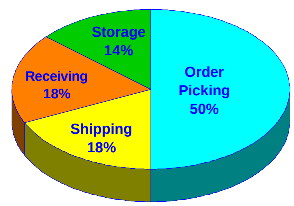
Overview

1. Introduction to Unit Load Storage Systems
2. Rack Storage Systems
3. Block Stacking Storage Systems
4. Storage Assignment Formulation & Algorithms
5. Conclusions

Warehousing Illustration



Warehousing Costs



Warehousing Objectives

- Minimize the Expected Travel Time
  - Minimize MH Equipment and Personnel

$$Min \sum_j f_j \cdot t_j$$

- Minimize the Required Storage Space
  - Minimize Capital Investment

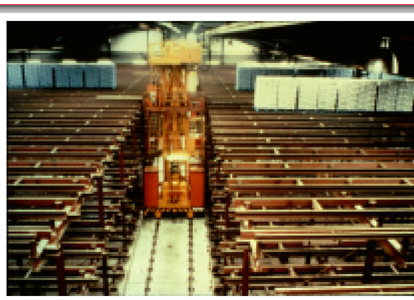
$$Min \;\; N$$

Pallet Rack Example

Multiple Aisles Automated Storage System Example

Block Stacking Storage Example

Deep Lane Storage Example



Overview

1. Introduction to Unit Load Storage Systems
2. **Rack Storage Systems**
3. Block Stacking Storage Systems
4. Storage Assignment Formulation & Algorithms
5. Conclusions

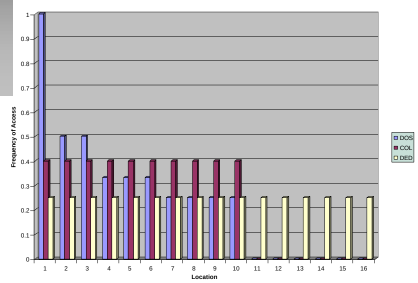
Unit Load Storage Policies

- No Information Policies
  - Random, Closest-Open-Location
- Product Characteristics Policies
  - Product Turnover Based
  - Class Product Turnover Based
- Item Characteristics Policies
  - Duration Of Stay
  - Zone Duration of Stay

Main Storage Principle to Minimize Travel

Place unit loads that generate the highest frequency of access in locations with the lowest expected distance

Shared Storage Example: Frequency of Access Distribution



Product Turnover Pure Dedicated

- Sort and Locate Products by Decreasing Turnover to Locations with Lowest Expected Distance

$$f_p = \frac{F_p}{q_p}$$

Product Turnover Class Based Storage

- Pure Dedicated is Very Space Inefficient
- 3 to 5 Classes based on Frequency of Access
- Dedicated Space for Each Class
- Class Space Determined by Simulation
- Inside Class Use Random or Closest Open Location
- Shared Storage Policy

Duration of Stay Shared Storage Observations

- Exploits that First and Last Unit Load in Batch are Different
- Cross Docking (DOS = 0)
- Minimizes Both Storage Space and Travel Time for a Perfectly Balanced Warehouse
- Very Constrained Perfectly Balanced Replenishment Pattern $n_p(t)$

Duration of Stay Shared Storage: Conclusions

- More Uncorrelated Products Favors DOS Policy
- More Balanced Warehouse Favors DOS Policy and Any Shared Policy

Storage Policy Conclusions

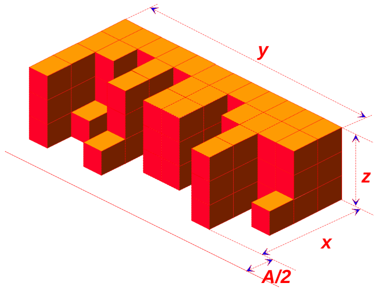
- Real Systems are Not Perfectly Balanced
- Duration of Stay Saves Travel and Storage Space
- 2 & 3 Class Product Performs Well
- Pure Turnover Dedicated Worst of All
- Savings Magnitude Depends on Replenishment Pattern
- DOS Data Requirements Indicate Warehouse Management System

Overview

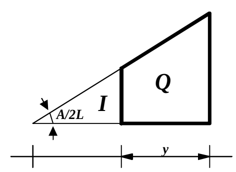
1. Introduction to Unit Load Storage Systems
2. Rack Storage Systems
3. Block Stacking Storage Systems
4. Storage Assignment Formulation & Algorithms
5. Conclusions

Block Stacking Storage System

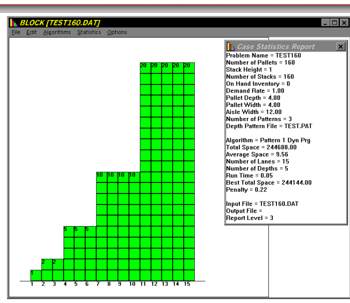
Block Stacking Illustration



Optimal Triangular Pattern



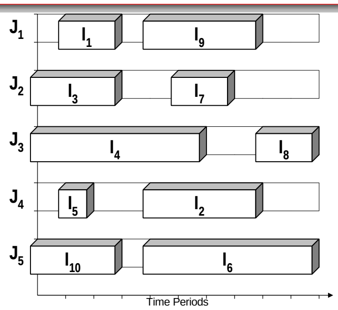
Block Stacking Design Program



Overview

1. Introduction to Unit Load Storage Systems
2. Rack Storage Systems
3. Block Stacking Storage Systems
4. Storage Assignment Formulation & Algorithms
5. Conclusions

Occupancy Gantt Chart



22-Jul-98 Unit Load Storage Systems Design Marc Goetschalckx

Unit Load Storage Formulation

$$\min \sum_{i=1}^{M}\sum_{j=1}^{N} c_{ij} x_{ij}$$

$$s.t. \quad \sum_{j=1}^{N} x_{ij} = 1 \qquad \forall i$$

$$\sum_{i=1}^{M} b_i x_{ij} \le 1 \qquad \forall j$$

$$x_{ij} \in \{0,1\}$$

22-Jul-98 Unit Load Storage Systems Design Marc Goetschalckx

Occupancy and Constraint Matrices

$$B = [b_i] = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

$$A = \begin{bmatrix} I & I & I & .. & I \\ B & 0 & 0 & .. & 0 \\ 0 & B & 0 & .. & 0 \\ .. & .. & .. & .. & .. \\ 0 & 0 & 0 & .. & B \end{bmatrix}$$

22-Jul-98 Unit Load Storage Systems Design Marc Goetschalckx

Vector Assignment Formulation

$$\max \sum_{i=1}^{M}\sum_{j=1}^{N} r_{ij} x_{ij}$$

$$r_{ij} = C - c_{ij} > 0 \qquad s.t. \quad \sum_{j=1}^{N} x_{ij} \le 1 \qquad \forall i$$

$$\sum_{i=1}^{M} b_i x_{ij} \le 1 \qquad \forall j$$

$$x_{ij} \in \{0,1\}$$

22-Jul-98 Unit Load Storage Systems Design Marc Goetschalckx

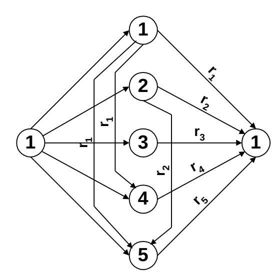
Vector Assignment Formulation

- Neither AP nor 3DAP
- Consecutive Ones in Occupancy Matrix B
- Block Diagonal Structure of Constraint Matrix
- Integrality Property Not Satisfied

22-Jul-98 Unit Load Storage Systems Design Marc Goetschalckx

Longest Path Subproblem

$$B = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$



22-Jul-98 Unit Load Storage Systems Design Marc Goetschalckx

Lagrangean Relaxation

$$\max \sum_{i=1}^{M}\sum_{j=1}^{N} r_{ij}x_{ij} + \sum_{i=1}^{M} m_i\left(1-\sum_{j=1}^{N} x_{ij}\right)$$

$$\max \sum_{i=1}^{M}\sum_{j=1}^{N} \left(r_{ij}-m_i\right)x_{ij} + \sum_{i=1}^{M} m_i$$

$$s.t. \quad \sum_{i=1}^{M} b_i x_{ij} \le 1 \qquad \forall j$$

$$x_{ij} \in \{0,1\}, m \ge 0$$

22-Jul-98 Unit Load Storage Systems Design Marc Goetschalckx

Computational Experiment

Activities	Resources	Heuristic	Upper Bound	% Gap	Optimal	Penalty
10	10	861	861	0.0	861	0.0
15	15	1357	1357	0.0	1357	0.0
30	9	2491	2491	0.0	2491	0.0
30	10	2631	2631	0.0	2631	0.0
30	15	2627	2627.1	0.0	2627	0.0
100	20	9024	9276.8	2.8	9270	-2.7
200	25	18869	18959.8	0.5		

22-Jul-98 Unit Load Storage Systems Design Marc Goetschalckx

VAP Conclusions

- Very Large Integer Problem
- Very Tight LP Relaxation
- Efficient Subproblem and Problem Size Indicate Decomposition
- Very Small Gap for Lagrangean Relaxation Upper Bound
- Acceptable Penalty for Primal Heuristic

22-Jul-98 Unit Load Storage Systems Design Marc Goetschalckx

Overview

1. Introduction to Unit Load Storage Systems
2. Rack Storage Systems
3. Block Stacking Storage Systems
4. Storage Assignment Formulation & Algorithms
5. Conclusions

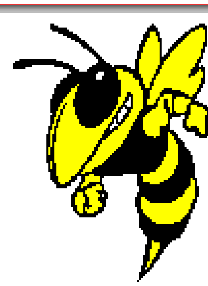
22-Jul-98 Unit Load Storage Systems Design Marc Goetschalckx

Unit Load Storage Systems: Conclusions

- Products are Not Uniform
- Duration-of-Stay of Individual Loads
- Fairly Simple Statistical Analysis Yields Large Savings
- Optimal, Mathematical Programming based Procedures for Small Systems

22-Jul-98 Unit Load Storage Systems Design Marc Goetschalckx

Thank You



22-Jul-98 Unit Load Storage Systems Design Marc Goetschalckx