Weekly presentation SCM research

9th presentation: Optimizing the Supplu Chain Configuration for New Products (Part 1)

Introduction

Authors

•Sean P. Willems

Associate Professor of Operations and Technology Management
Boston University School of Management

•Stephen C. Graves

Interim Director, MIT Engineering Systems Division
Abraham J. Siegel Professor of Management Science
Professor of Engineering Systems and Mechanical Engineering

Introduction

Papers

•*Supply Chain Configuration and Part Selection in Multigeneration Products*, Massachusetts Institute of Technology (1999)

•*Optimizing the Supply Chain Configuration for New Products*, Management Science (2005)

Summary

I/ Context

1) Introduction

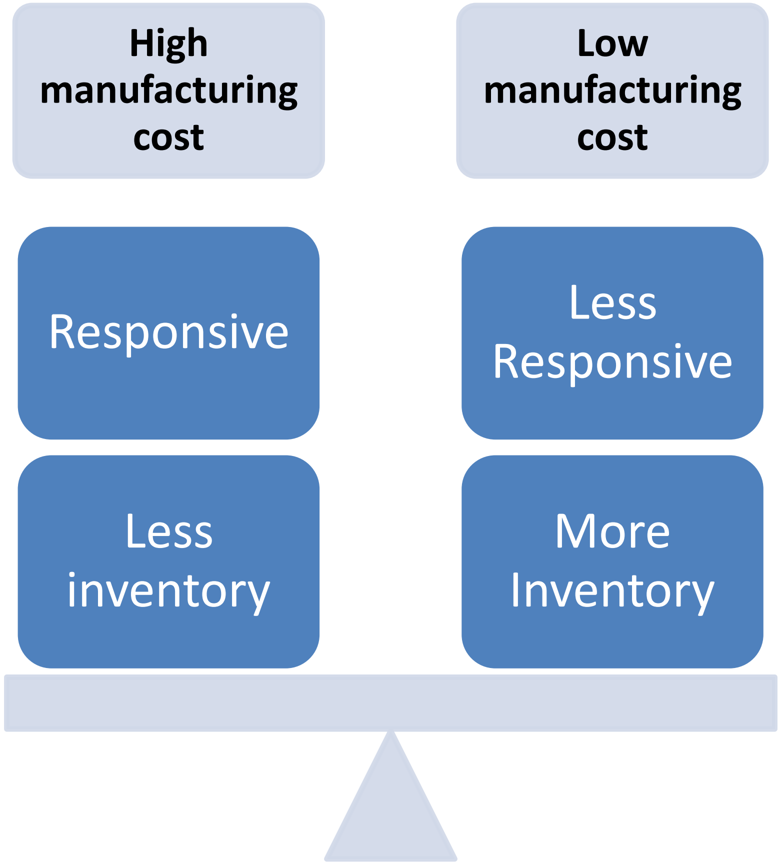
- Intent
  - Develop support decision tool to use during product development process where the product design has been fixed,
  - Vendors, manufacturing technologies, shipments options to be decided

- Model
  - A Supply Chain can be viewed as a network where the nodes represent functionnality that must be provided and the arcs the constraints among the functions
  - Several options for each functions, characterized by their **direct cost** and **lead time**

Objective is to identify the options that can satisfy each function and then select the best one that ***minimize the entire Supply Chain cost***

I/ Context
1) Introduction

- Tradeoff



I/ Context
1) Introduction

UMC (Unit Manufacturing Cost) is the dominant criterion in the Supply Chain design

- Gross margin target
- Esay to calculate directly

- 3 specific costs to minimize

Safety Stock	Pipeline Stock	COGS
• *Expected inventory in oder not to fall in shortage*	• *Work-in-process inventory*	• *Cost Of Goods Sold* • *Total cost of all the units that are delivered to customers during a company-defined period of time*

I/ Context

2) Notation and Assumptions

•Supply Chain seen as a network of stages, with several options to choose

•Lead Time is the time to perform the function at the stage, provided all the inputs are available

•Direct Cost represents the direct material and direct labor costs associated with the option

•Periodic review policy

•Base-Stock replenishment policy

•No time delay in ordering

I/ Context
2) Notation and Assumptions

- Demand process

  - External demand occurs only at noeds with no successors, stationary process with average demand per period μ_j

  - Internal stage

$$d_i(t) = \sum_{(i,j)\epsilon A} d_j(t) \qquad \mu_i(t) = \sum_{(i,j)\epsilon A} \mu_j(t)$$

  - Demand is bounded by the function Dj, which is increasing and concave

I/ Context
2) Notation and Assumptions

- Guranteed service time

  - Outbound service time s_i^{out} by which the stage will satisfy its demand with 100% service

  - Maximum service time Sj

  - Inbound service time service s_i^{in} time to receive all the required inputs from its suppliers

II/ Optimization Model

1) Inventory model

•Ii(t), the finished inventory at stage i at the end of period t

$$I_i(t) = B_i - d_i(t - s_i^{in} - t_i, t - s_i^{out})$$

•Bi=Ii(0)

• $d_i(t - s_i^{in} - t_i, t - s_i^{out})$ represents the inventory shortfall between the cumulative replenishment and the cumulative shipment

•In order to gurantee 100% service, $B_i = D_i(\tau)$
with $\tau = \max\{0; s_i^{in} + t_i - s_i^{out})$

• the demand over the net replenishment time is demand that has been filled but has not yet been replenished

•If the replenishment time is negative then there is no need for inventory

II/ Optimization Model
1) Inventory model

- Safety Stock

$$SS_i = D_i(s_i^{in} + t_i - s_i^{out}) - (s_i^{in} + t_i - s_i^{out})\mu_i$$

- Pipeline inventory

$$PI_i(t) = t_i\mu_i$$

II/ Optimization Model
2) Mathematical formulation

$$\mathbf{P}\quad \min \sum_{i=1}^{N} \alpha c_i \left[D_i\left(s_i^{in} + t_i - s_i^{out}\right) - \left(s_i^{in} + t_i - s_i^{out}\right)\mu_i \right]$$

Safety stock

$$+\ \alpha\left(c_i - \frac{x_i}{2}\right) t_i \mu_i + \beta x_i \mu_i$$

Pipeline inventory *COGS*

•Where

- •*Di() = maximum demand function for stage i*
- •*α = scalar representing the holding cost rate*
- •*β = scalar converting the model's underlying time unit into the company's time interval of interest (typically 1 year)*
- •*μi = mean demand rate at stage i*
- •*ci = cumulative cost at stage i*
- •*ti = selected option's lead time at stage i*
- •*xi = selected option's cost at stage i*

II/ Optimization Model
2) Mathematical formulation

- Constraints
  - *Stage cost and lead time*

$$\sum_{i=1}^{O_i} T_{ik} y_{ik} - t_i \quad for\ i = 1, 2, \dots, N \qquad \sum_{i=1}^{O_i} C_{ik} y_{ik} - x_i \quad for\ i = 1, 2, \dots, N$$

  - *Stage cumulative cost*

$$c_i - \sum_{j:(i,j)\epsilon A} c_j - x_i \quad for\ i = 1, 2, \dots, N$$

  - *Service times*

$$s_i^{in} \geq s_j^{out} \quad for\ i = 1, 2, \dots, N; j: (i, j) \in A$$

$$s_i^{in} + t_i - s_i^{out} \geq 0 \quad for\ i = 1, 2, \dots, N$$

$$s_j^{out} \leq S_j \quad for\ all\ demand\ nodes\ j$$

  - *Options sourcing*

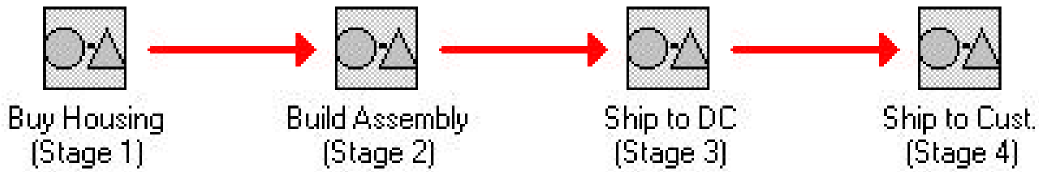
$$\sum_{k=1}^{O_i} y_{ik} = 1 \quad for\ i = 1, 2, \dots, N$$

III/ Dynamic programming
1) Special SC cases

- Programming pattern
  - *Identify the appropriate network model for the Supply Chain*
  - *Define the different subgraphes to solve the problem step by step*
  - *Define the supply chain cost for the subnetwork*
  - *Deduct the functional equation (minimum supply chain cost) according to the constraints*
  - *Solve using forward recursion*

III/ Dynamic programming
1) Special SC cases

•Serial line

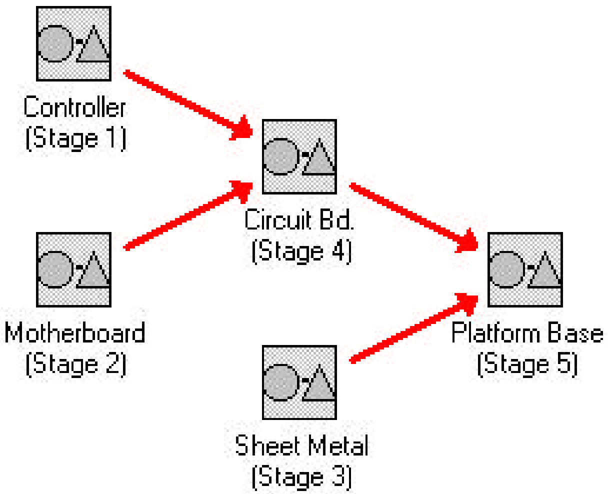


•Only upstream cost included

•First evaluate fN(c,SN) then solve backtrack

III/ Dynamic programming
1) Special SC cases

•Assembly Network



•*Severals suppliers but only one downstream stage*

•*Topologically ordered*

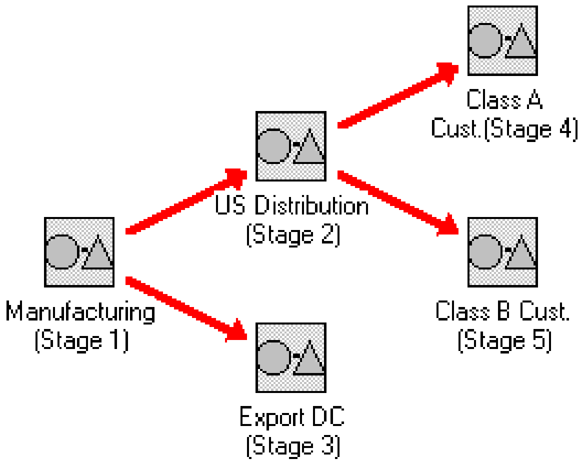
•*One cumulative cost can represent different configurations*

•*CI = incoming cumulative cost*

•*Evaluate all combinations for each value of CI*

III/ Dynamic programming
1) Special SC cases

•Distribution Network



•Several customers but only one upstream stage

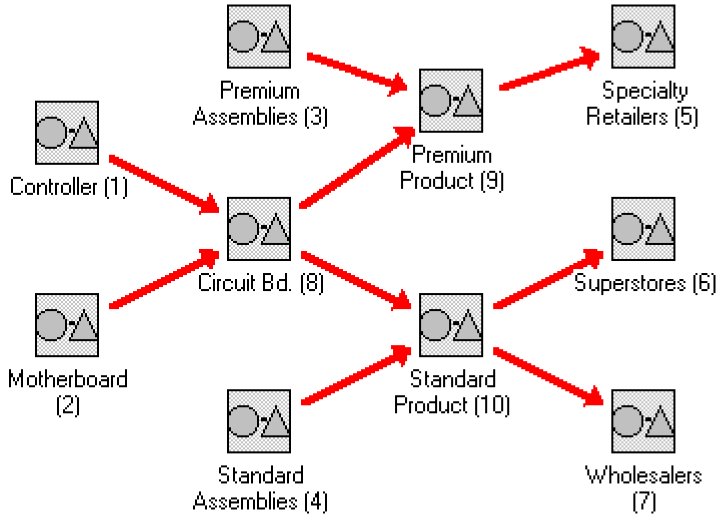
•Same service time for all customers

•Start solving at Fi(0,0) then forward

III/ Dynamic programming
2) Spanning tree

- Spanning tree = connected graph that contains N nodes and N-1 arcs
- Represent numerous kinds of real world supply chains

  Common component that goes into different final assemblies that have each different distribution channels



III/ Dynamic programming
2) Spanning tree

- Nodes are labeled such that for each node i there is at most one adjacent node with a higher number, called i's parent, p(i)
- Ni, subset of nodes {1,2,…,i} that are connected to i

$$N_i = \{i\} + \bigcup_{h<i,(h,i)\in A} N_h + \bigcup_{j<i,(i,j)\in A} N_j$$

- 2 forms of the functional equation
- Minimum cost for the supply chain configuration in a subnetwork with node set Ni
  - When p(i) is downstream, $f_i(c^T, s^{out})$
  - When p(i) is upstream, $g_i(c^1, s^{in})$

III/ Dynamic programming
2) *Spanning tree*

•Supply chain cost for the subnetwork with node set Ni when option k is selected for stage i

$$z_{ik}\left(s^{in}, c^1, c^2, s^{out}\right) =$$

$$\alpha c^T\left[D_i\left(s_i^{in} + T_{ik} - s_i^{out}\right) - \left(s_i^{in} + T_{ik} - s_i^{out}\right)\mu_i\right]$$

Safety stock

$$+\,\alpha\left(c^T - \frac{C_{ik}}{2}\right)T_{ik}\mu_i \quad +\beta C_{ik}\mu_i + \sum_{\{j:(i,j)\in A, j<i} g_j(c^T, s^{out}) +$$

Pipeline inventory *COGS* *downstream*

$$\min_{\sum_{\{h:(h,i)\in A, h<i\}} c_{h=c^2}} \left\{ \sum_{\{h:(h,i)\in A, h<i\}} f_h(c_h, s^{in}) \right)$$

upstream

III/ Dynamic programming

2) Spanning tree

•Dynamic Program

For i=1 to N-1

- •If p(i) is downstream of i, for all feasible values of the variables evaluate

$$f_i(c^T, s^{out}) = \min_{k,s^{in}}\{z_{ik}(s^{in}, 0, c^2, s^{out})\}$$

- •If p(i) is upstream of i, evaluate

$$g_i(c^1, s^{in}) = \min_{k,c^2,s^{out}}\{z_{ik}(s^{in}, c^1, c^2, s^{out})\}$$

•Minimize gN(0,Sin) for all Sin feasible to obtain the optimal objective function value

•Computational complexity is of order $k^N N M^2$

Thank you for your attention!