Quantitative Models for Supply Chain Design and Management

Raul Poler

Valencia
Universidad Politécnica de Valencia
Camino de Vera s/n, Ed. 8G - Esc. 4 - Nivel 1 y 4 (Ciudad Politécnica de la Innovación)
46022 Valencia - España

Alcoy
Esc. Politécnica Superior de Alcoy
Universidad Politécnica de Valencia
Plaza Ferrándiz Carbonell nº2, Edificio
03801 Alcoy (Alicante) - España

Tel.: (34) 963 879 680 - Fax: (34) 963 879 682 - e-mail: info@cigip.org
Profile

Raul Poler
PhD Industrial Engineering
Professor at Polytechnic University of Valencia (UPV)
Director of Research Centre on Production Management and Engineering (CIGIP)
Founder partner of Spin-off UPV EXOS Solutions
Director of Master in Industrial Engineering and Logistics (MUIOL)
LinkedIn
Google Scholar
Outline

- Introduction
- Mathematical Programming Models
- Strategic Models for Supply Chain Design
  - The single-source facility location problem
  - The distribution system problem
  - The integrated production/distribution system problem
- Tactical Models for Supply Chain Planning
  - The integrated production/distribution planning problem
  - The multisite supply chains planning problem
- Operational Models for Supply Chain Scheduling
  - The processing unit scheduling problem
Outline

- Introduction
- Mathematical Programming Models
- Strategic Models for Supply Chain Design
  - The single-source facility location problem
  - The distribution system problem
  - The integrated production/distribution system problem
- Tactical Models for Supply Chain Planning
  - The integrated production/distribution planning problem
  - The multisite supply chains planning problem
- Operational Models for Supply Chain Scheduling
  - The processing unit scheduling problem
Introduction

- The Supply Chain planning matrix

Introduction

The Supply Chain planning matrix

- Strategic Level
  - Supply Chain Design
- Tactical Level
  - SC Operations Planning
- Operational Level
  - SC Operations Scheduling
Introduction

Quantitative Models for SCM – Taxonomy:

- **Decision level**
  - Strategic ✓
  - Tactical ✓
  - Operational ✓

- **Modelling approach**
  - Deterministic ✓
  - Stochastic
  - Hybrid
  - IT-driven

- **Physical environment**
  - One-stage ✓
  - Two-stage ✓
  - Multi-stage ✓
Introduction

Quantitative Models for SCM – Taxonomy:

- **Decision level**
  - **Strategic**:
    - Horizon of 5-10 years
    - It affects long-term system performance
    - System Design
    - Resource Acquisition
  - **Tactical**:
    - Horizon 1-2 years
    - Medium term affects system performance
    - Deciding on the best use of different acquired resources
  - **Operational**:
    - 1-18 months horizon
    - Very small time periods
    - Sequencing and timing of operations
Introduction

- Quantitative Models for SCM – Taxonomy:
  - Physical environment
    - Stages
      - One-stage, two-stage, multi-stage
    - Stages considered in the Supply Chain structure:
      - Component Supplier (CS)
      - Inbound Logistics (IN)
      - Assembly Plants (AP)
      - Outbound Logistics (OUT)

Introduction

- Quantitative Models for SCM – Taxonomy:
  - Physical environment
    - Layers


Quantitative Models for Supply Chain Design and Management – Raul Poler
Introduction

Quantitative Models for SCM – Taxonomy:

- **Modelling approach**
  - Deterministic models
    - All model parameters are fixed and known with certainty
  - Stochastic models
    - Some model parameters are uncertain or random
  - Hybrid models
    - Mixed models (deterministic and stochastic)
  - IT-driven models
    - Try to integrate and coordinate various stages of Supply Chain planning on a real time basis and to improve visibility through the Supply Chain using software applications

Introduction

Quantitative Models for SCM – Taxonomy:

- **Modelling approach**
  - Deterministic models
    - Single objective
    - Multiple objectives
  - Stochastic models
    - Optimal Control Theory
    - Dynamic Programming
  - Hybrid models
    - Inventory Theoretic
    - Simulation
  - IT-driven models
    - CPFR (Collaborative Planning and Forecasting Replenishment)
    - MRP (Materials Requirement Planning)
    - DRP (Distribution Resource Planning)
    - ERP (Enterprise Resource Planning)
    - GIS (Geographic Information System)
Introduction

- Quantitative Models for SCM: Centralized vs Distributed

  **Centralized decision making**
  - A single decision-maker that has authority to manage the operations of all entities of the Supply Chain
    - Centralizes all the needed information necessary for decision making
    - Takes decisions on the proper operation of the SC as a whole
    - Based on some agreed objectives of the SC partners
    - A single quantitative model

  **Distributed decision making**
  - Several decision-makers
    - Each decider makes its own plans based on:
      - Own objectives and constrains
      - Some private information
    - As much quantitative models as as decision-makers
    - Need for coordination of all quantitative models

*Which will achieve the optimum for all the Supply Chain?*
Outline

- Introduction
- Mathematical Programming Models
  - Strategic Models for Supply Chain Design
    - The single-source facility location problem
    - The distribution system problem
    - The integrated production/distribution system problem
  - Tactical Models for Supply Chain Planning
    - The integrated production/distribution planning problem
    - The multisite supply chains planning problem
  - Operational Models for Supply Chain Scheduling
    - The processing unit scheduling problem
Mathematical Programming Models

Mathematical Programming

- In Operations Research, Mathematical Programming is the selection of a best element (with regard to some criteria) from some set of available alternatives.

Main types:

- Linear Programming (LP)
  - Continuous Linear Programming (CLP)
  - Integer Linear Programming (ILP)
  - Mixed Integer Linear Programming (MILP)
- Non-Linear Programming (NLP)
  - Continuous Non-Linear Programming (CNLP)
  - Integer Non-Linear Programming (INLP)
  - Mixed Integer Non-Linear Programming (MINLP)
- Quadratic Programming (QP)
Mathematical Programming Models

- **Structure**
  - **Indexes**
    - Model elements (factory, machine, part, etc.)
  - **Sets**
    - Groups of instances of indexes (set of machines, etc.)
  - **Parameters, data**
    - Known attributes which can not be changed (cost, demand, etc.)
  - **Decision variables**
    - Unknown attributes which can be changed (amount to produce, etc.)
  - **Objective(s)**
    - Value(s) which the decision-maker wants to optimize (maximize or minimize)
  - **Constraints**
    - Limits to satisfy (resources capacity, available money, etc.)
A simple example

Nitron Corporation manufactures 2 products (A and B) using 2 machines (P and Q). Product A provides a benefit of 60€ per unit, product B provides a benefit of 50€ per unit. Each unit of product A requires 10 min of machine P and 8 min of machine Q. Each unit of product B requires 20 min of machine P and 5 min of machine Q. Machine P capacity is 200 min per day. Machine Q capacity is 80 min per day. The minimum production should be 2 units of A and 5 units of B per day. ¿How many units A and B should Nitron produce per day?
Mathematical Programming Models

- Indexes:
  - $i$: products
  - $j$: machines

- Sets:
  - $i$: \{A, B\}
  - $j$: \{P, Q\}

- Data:
  - $Be_i$: [60, 50] $\rightarrow$ $Be_A = 60 ; Be_B = 50$
  - $Mp_i$: [2, 5] $\rightarrow$ $Mp_A = 2 ; Mp_B = 5$
  - $Ca_j$: [200, 80] $\rightarrow$ $Ca_P = 200 ; Ca_Q = 80$
  - $Req_{ij}$: [[10, 8] [20, 5]] $\rightarrow$ $Req_{AP} = 10 ; Req_{AQ} = 8 ; Req_{BP} = 20 ; Req_{BQ} = 5$

- Decision variables:
  - $X_i \rightarrow X_A ; X_B$
Mathematical Programming Models

- **Objective:**
  Maximize benefit

  \[ MaxZ = \sum_i Be_i \cdot X_i \]

- **Constraints:**
  Machines capacity

  \[ \sum_i Req_{ij} \cdot X_i \leq Ca_j \quad j = \{P, Q\} \]

  Minimum production

  \[ X_i \geq Mp_i \quad i = \{A, B\} \]
Mathematical Programming Models

- Independency of data
- Solve different problems when data changes
- Scalability
  - 2 products → 100 products
  - 2 machines → 50 machines
Mathematical Programming Models

- Data stored in a DATABASE

![Database Diagram]
Mathematical Programming Models

- Model written with an Algebraic Modelling Language (AML)
  - AMPL
    - [http://AMPL.com/](http://AMPL.com/)
  - GAMS (General Algebraic Modelling System)
    - [http://GAMS.com/](http://GAMS.com/)
  - LINDO/LINGO
  - MPL (Mathematical Programming Language)
    - [http://maximalsoftware.com/](http://maximalsoftware.com/)
  - Pyomo
    - [http://www.pyomo.org/](http://www.pyomo.org/)
  - JuMP (Julia for Mathematical Programming)
Mathematical Programming Models

Classwork → model in MPL

- Create the tables in the Nitron.mdb database and fill the data
- Create the Nitron.mpl model
- Obtain the solution
Mathematical Programming Models

- Model sections in MPL

  TITLE

  OPTIONS

  INDEX

  DATA

  VARIABLES

  MACROS

  MODEL

  SUBJECT TO

  BOUNDS

  INTEGER

  BINARY

  FREE

  END
Mathematical Programming Models

Model in MPL

! Nitron Corporation
TITLE
   Nitron;
OPTIONS
   DatabaseType=Access;
   DatabaseAccess="Nitron.mdb";
INDEX
   i := DATABASE("Products",  "IdProduct");
   j := DATABASE("Machines",  "IdMachine");
DATA
   Be[i] := DATABASE("Products",  "Benefit");
   Mp[i] := DATABASE("Products",  "MinProduction");
   Ca[j] := DATABASE("Machines",  "Capacity");
   Req[i,j] := DATABASE("Requirements",  "Requirement");
VARIABLES
   X[i]     EXPORT TO DATABASE("Products" ,  "Production");
MACROS
   Benefit := SUM(i: Be[i]*X[i]);
MODEL
   MAX Z = Benefit;
SUBJECT TO
   RCa[j]  : SUM(i:Req[i,j]*X[i]) <= Ca[j];
BOUNDS
   X[i] >= Mp[i];
END
Mathematical Programming Models

Model check in MPL

![Image of MPL software interface showing a mathematical model for Nitron Corporation]

- **TITLE**: Nitron
- **OPTIONS**: Database Type = Access, Database Access = "Nitron.mdb"
- **INDEX**: i, j
- **DATA**: Be[i], Mp[i], Ca[j], Req[i,j]
- **VARIABLES**: X[i]
- **MACROS**: Benefit := SUM(i: Be[i] * X[i])
- **MODEL**: MAX Z = Benefit; SUBJECT TO: RCA[j] := SUM(i: Req[i,j] * X[i]) <= 0; BOUNDS: X[i] >= Mp[i]

**Status Window**
- The syntax of 'Nitron.mpl' is correct.
- Main File: Nitron.mpl, Lines: 25, Time: 0.067s
- Model: Variables: 0, Nonzeros: 0, Constraints: 0, Integers: 0
- Solver: Iterations, Objective Function
  - Phase1: 0, 0.0000
  - Total: 0, 0.0000

Check Syntax
Mathematical Programming Models

Model solve in MPL

```plaintext
TITLE
Nitron;
OPTIONS
    DatabaseType=Access;
    DatabaseAccess="Nitron.mdb";
INDEX
    i   := DATABASE("Products", "E1");
    j   := DATABASE("Machines", "E1");
DATA
    B[i]  := DATABASE("Products", "E1");
    M[i]  := DATABASE("Products", "E1");
    C[j]  := DATABASE("Machines", "E1");
    R[i,j] := DATABASE("Requirements", "E1");
VARIABLES
    X[i]  EXPORT TO DATABASE("Products", "E1");
MACROS
    Benefit := SUM(i: B[i] * X[i]);
MODEL
    MAX Z = Benefit;
SUBJECT TO
    R[j]  := SUM(i: R[i,j] * X[i]) 
        <= C[j];
    B[i]  >= M[i];
END
```

Status Window

Optimal solution found

<table>
<thead>
<tr>
<th>Model</th>
<th>Variables</th>
<th>Nonzeros</th>
<th>Constraints</th>
<th>Integers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nitron.mpl</td>
<td>2</td>
<td>6</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

Solver

<table>
<thead>
<tr>
<th>Iterations</th>
<th>Objective Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.0000</td>
</tr>
<tr>
<td>2</td>
<td>709.0003</td>
</tr>
</tbody>
</table>

Solve
Mathematical Programming Models

- **MPL solution**

```plaintext
SOLUTION RESULT
Optimal solution found
MAX Z = 709.0909

MACROS
Macro Name | Values
--- | ---
Benefit | 709.0909

DECISION VARIABLES
VARIABLE X[i] :

<table>
<thead>
<tr>
<th>i</th>
<th>Activity</th>
<th>Reduced Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>5.4545</td>
<td>0.0000</td>
</tr>
<tr>
<td>B</td>
<td>7.2727</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

CONSTRAINTS
CONSTRAINT RCa[j] :

<table>
<thead>
<tr>
<th>j</th>
<th>Slack</th>
<th>Shadow Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>0.0000</td>
<td>2.0909</td>
</tr>
<tr>
<td>Q</td>
<td>0.0000</td>
<td>3.6364</td>
</tr>
</tbody>
</table>

END
```
Mathematical Programming Models

- Solution in database

<table>
<thead>
<tr>
<th>IdProduct</th>
<th>Benefit</th>
<th>MinProduction</th>
<th>Production</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>50</td>
<td>2</td>
<td>5,4545454545</td>
</tr>
<tr>
<td>B</td>
<td>60</td>
<td>5</td>
<td>7,2727272727</td>
</tr>
</tbody>
</table>
Mathematical Programming Models

- Model in MPL (integer values)

```
TITLE
   Nitron;
OPTIONS
   DatabaseType=Access;
   DatabaseAccess="Nitron.mdb";
INDEX
   i       := DATABASE("Products", "IdProduct");
   j       := DATABASE("Machines", "IdMachine");
DATA
   Be[i]   := DATABASE("Products", "Benefit");
   Mp[i]   := DATABASE("Products", "MinProduction");
   Ca[j]   := DATABASE("Machines", "Capacity");
   Req[i,j]:= DATABASE("Requirements", "Requirement");
VARIABLES
   X[i]     EXPORT TO DATABASE("Products", "Production");
MACROS
   Benefit := SUM(i: Be[i]*X[i]);
MODEL
   MAX Z = Benefit;
SUBJECT TO
   RCa[j] : SUM(i:Req[i,j]*X[i]) <= Ca[j];
BOUNDS
   X[i] >= Mp[i];
INTEGER
   X[i];
END
```
Mathematical Programming Models

- MPL solution (integer values)

SOLUTION RESULT
Optimal integer solution found
MAX Z = 680.0000

MACROS
Macro Name Values
-----------------------------------------------
Benefit 680.0000

DECISION VARIABLES
VARIABLE X[i] :
i Activity Reduced Cost
---------------------------------------
A 4.0000 0.0000
B 8.0000 -40.0000

CONSTRAINTS
CONSTRAINT RCa[j] :
j Slack Shadow Price
---------------------------------------
P 0.0000 5.0000
Q 8.0000 0.0000

END
Mathematical Programming Models

Solution in database (integer values)

<table>
<thead>
<tr>
<th>IdProduct</th>
<th>Benefit</th>
<th>MinProduction</th>
<th>Production</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>50</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>B</td>
<td>60</td>
<td>5</td>
<td>8</td>
</tr>
</tbody>
</table>
Outline

- Introduction
- Mathematical Programming Models
- Strategic Models for Supply Chain Design
  - The single-source facility location problem
  - The distribution system problem
  - The integrated production/distribution system problem
- Tactical Models for Supply Chain Planning
  - The integrated production/distribution planning problem
  - The multisite supply chains planning problem
- Operational Models for Supply Chain Scheduling
  - The processing unit scheduling problem
The single-source facility location problem

- **Strategic / One-stage / Deterministic**
  - **Goal:** locating a set of warehouses in a distribution network
    - Retailers geographically dispersed in a region
    - There are \( m \) preselected locations as possible store locations
    - Retailers want to receive products from a single warehouse
  - **The cost of placing a warehouse at a particular location includes**
    - Fixed Cost: construction costs, maintenance, etc.
    - Variable costs: transport costs
  - **Decision variable:** locations where to locate the warehouses
  - **Objective:** minimize the total cost

The single-source facility location problem

- warehouses (unknown design)
- customers (known design)
Strategic Models for Supply Chain Design

- Indexes:
  - $i$: retailers
  - $j$: locations (in which place the warehouses)

- Data:
  - $d_i$: yearly demand from retailer $i$
  - $b_{ij}$: cost of transporting $d_i$ units from warehouse $j$ to retailer $i$
  - $F_j$: yearly operation cost of warehouse $j$
  - $q_j$: capacity of warehouse $j$ in units

- Decision variables:
  - $Y_j$: binary { 1 if a warehouse is placed in location $j$ } { 0 otherwise }
  - $X_{ij}$: binary { 1 if the warehouse $j$ supplies the retailer $i$ } { 0 otherwise }

The single-source facility location problem
Strategic Models for Supply Chain Design

- **Objective:**
  
  Minimize transport cost and operation cost

  \[ \text{Min} Z = \sum_i \sum_j b_{ij} \cdot X_{ij} + \sum_j F_j \cdot Y_j \]

- **Constraints:**

  Units transported from a warehouse \(j\) to all the retailers \(i\) (to which it supplies) should be less than its capacity

  \[ \sum_i d_i \cdot X_{ij} \leq q_j \cdot Y_j \quad \forall j \]

  Retailers should receive products from a single warehouse

  \[ \sum_j X_{ij} \leq 1 \quad \forall i \]
Strategic Models for Supply Chain Design

Constraints: (cont)

Demand from all the retailers \( i \) should be fulfilled

\[
\sum_{j} d_i \cdot X_{ij} \geq d_i \quad \forall i
\]
Strategic Models for Supply Chain Design

Classwork → model in MPL

- Use the SSFLP.mdb database
- Create the SSFLP.mpl model
- Obtain the solution

The single-source facility location problem
Outline

- Introduction
- Mathematical Programming Models
- **Strategic Models for Supply Chain Design**
  - The single-source facility location problem
  - The distribution system problem
  - The integrated production/distribution system problem
- **Tactical Models for Supply Chain Planning**
  - The integrated production/distribution planning problem
  - The multisite supply chains planning problem
- **Operational Models for Supply Chain Scheduling**
  - The processing unit scheduling problem
The distribution system problem

Strategic / One-stage / Deterministic

Goal: define the optimum network of distribution warehouses for distributing products to retailers (regions) from production plants
- Production plants are known (amount and location)
- Retailers are known and grouped in regions
- Warehouses should be built in pertinent locations

Costs:
- Fixed Cost: warehouses construction; warehouses operation
- Variable costs: transport costs between production plants and warehouses and between warehouses and regions; warehouses maintenance

Decision variables:
- Locations where to locate the warehouses; warehouses assignment to regions; amount of products transported from plants to warehouses and from warehouses to regions

Objective: minimize the total cost
The distribution system problem

plants (known design)  warehouses (unknown design)  regions (known design)

Strategic Models for Supply Chain Design

- **Indexes:**
  - $i$: production plants
  - $j$: warehouses
  - $k$: regions (of retailers)
  - $l$: products

- **Data:**
  - $d_{kl}$: demand from region $k$ of product $l$
  - $a_{ijl}$: cost of transporting 1 unit of product $l$ from plant $i$ to warehouse $j$
  - $b_{jkl}$: cost of transporting 1 unit of product $l$ from warehouse $j$ to region $k$
  - $I_j$: cost of building warehouse $j$
  - $F_j$: yearly operation cost of warehouse $j$
  - $v_{jl}$: handling cost of 1 unit of product $l$ in warehouse $j$
  - $c_{il}$: yearly production capacity of product $l$ in plant $i$
  - $C$: maximum amount of warehouses to build
  - $B$: maximum investment for warehouses building
Strategic Models for Supply Chain Design

- Decision variables:
  - $Y_j$: binary { 1 if a warehouse $j$ is built} { 0 otherwise }
  - $W_{jk}$: binary { 1 if the warehouse $j$ supplies the region $k$ } { 0 otherwise }
  - $S_{ijl}$: units of product $l$ transported from plant $i$ to warehouse $j$
  - $T_{jkl}$: units of product $l$ transported from warehouse $j$ to region $k$

- Objective:
  Minimize transport cost, handling cost and building and operation cost

$$MinZ = \sum_{i} \sum_{j} \sum_{l} a_{ijl} \cdot S_{ijl} + \sum_{j} \sum_{k} \sum_{l} b_{jkl} \cdot T_{jkl} + \sum_{j} \sum_{k} \sum_{l} d_{kl} \cdot v_{jl} \cdot W_{jk} + \sum_{j} \left(F_j \cdot Y_j + I_j \cdot Y_j\right)$$
Strategic Models for Supply Chain Design

Constraints:

Material flows

The demand of all products from all regions should be satisfied

$$\sum_{j} T_{jkl} = d_{kl} \quad \forall k, \forall l$$

The amount of each product which arrives to a warehouse should be equal to the amount which exit from that warehouse

$$\sum_{i} S_{ijl} = \sum_{k} T_{jkl} \quad \forall j, \forall l$$

Physical resources limitations

The amount of each product produced by a plant should not exceed the production capacity

$$\sum_{j} S_{ijl} \leq c_{il} \quad \forall i, \forall l$$
The distribution system problem

Constraints: (cont)

Financial resources limitations
The amount of money invested in building warehouses should not exceed the investment budget

\[ \sum_{j} I_j \cdot Y_j \leq B \]

Company policies
Each region receives all its products from only one warehouse

\[ \sum_{j} W_{jk} = 1 \quad \forall k \]

The number of warehouses built should not exceed the limit

\[ \sum_{j} Y_j \leq C \]
Constraints: (cont)

Logical constraints

A warehouse will supply a region only when the amount transported of all products from such warehouse to such region is nonzero

$$\sum_{l} T_{jkl} \leq M \cdot W_{jk} \quad \forall j, \forall k$$

A warehouse should be built if it supplies to any region

$$W_{jk} \leq Y_{j} \quad \forall j, \forall k$$
Strategic Models for Supply Chain Design

Classwork → model in MPL

- Use the DSP.mdb database
- Create the DSP.mpl model
- Obtain the solution