The Stochastic Warehouse-Inventory-Transportation Problem:

A Branch-and-Bound Method for Stochastic Integer Bilinearly-Constrained Programs

Christopher Hagmann, Dr. Nan Kong

Purdue University

Joint work with Dr. Parikh and Dr. Sainathuni

November 10, 2014

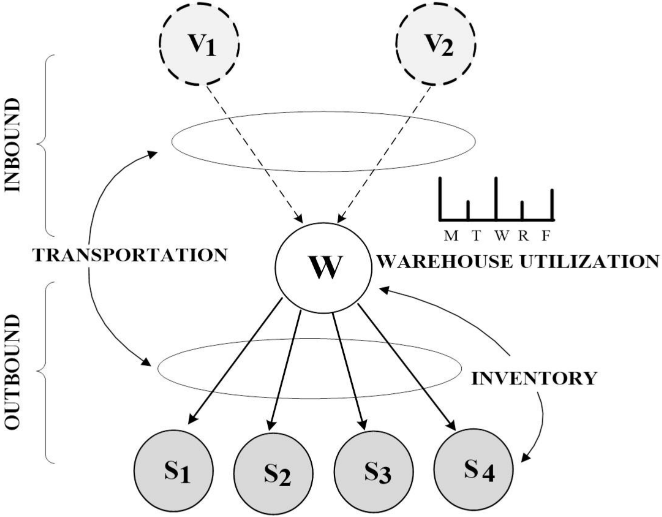
PURDUE UNIVERSITY

WRIGHT STATE UNIVERSITY

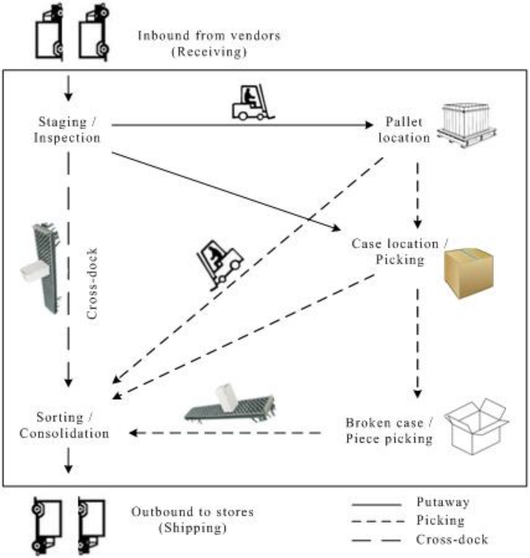
NSF

CMMI #1235283 and #1235061

Two echelon supply-chain model



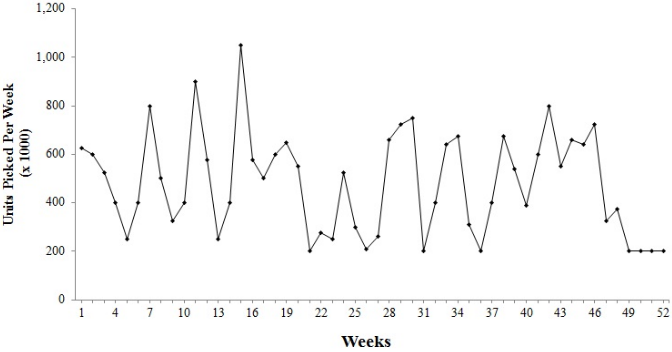
Common warehousing activities



Workload Variation

Weekly workload variation in a US-based apparel supply chain

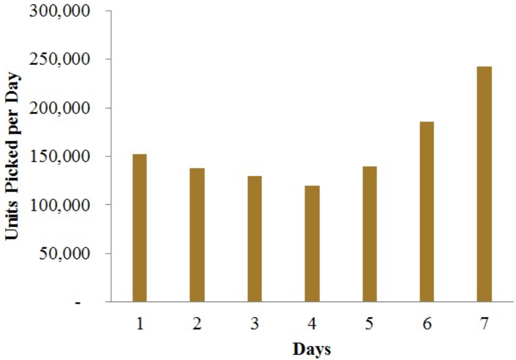
- Period: January 2011 - December 2011
- 42 - 219% variation in warehouse workload



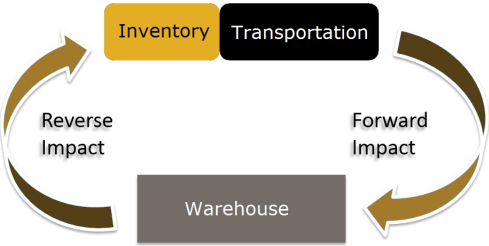
Workload Variation

Daily workload variation in the US warehouse of a Fortune 500 Grocery Distributor

- Period: August 29, 2011 - September 4, 2011
- 76 - 153% variation in warehouse workload



Incorporating Warehousing with Inventory and Transportation



Impact of
- Technology used
- Workforce level

on
- Shipment schedules and quantity
- Inventory at warehouse and stores

Impact of
- Shipment schedules
- Shipment quantity
- Inventory levels

on
- Warehouse workload
- Workforce planning

Research Objectives

Research Objectives

The purpose of this research is:

- To determine if proactively incorporating warehousing decisions during the planning stage would advance the way supply chains are designed and operated in an uncertain environment.
- To develop methods to solve two-stage stochastic integer bilinearly-constrained programing problem (SIBCP).

Incorporating Uncertainty

Assumptions

- Demand is the only uncertainty source.
- Full-time employees cannot be hired and fired with every time step.
- Technology is expensive and cannot be bought with every time step.

These motivate the need of a two-stage stochastic problem, with full-time workforce level and technology usage decisions in the first-stage.

First Stage Variables

$$
\begin{aligned}
\textit{Minimize} \quad & C^{\alpha}(\alpha_1+\alpha_2)+\sum_i C_i^{\theta_1}\theta_{1i}+\sum_j C_j^{\theta_2}\theta_{2j}+\sum_k p_k\phi(\alpha,\lambda,k) && (1)\\
\textit{s.t.} \quad & \sum_i \theta_{1i}=1; \sum_j \theta_{2j}=1 && (2)\\
& \sum_i \Lambda_{1i}\theta_{1i}=\lambda_1; \sum_j \Lambda_{2j}\theta_{2j}=\lambda_2 && (3)\\
& \gamma A_1 \le \alpha_1\lambda_1 && (4)\\
& \gamma A_2 \le \alpha_2\lambda_2 && (5)\\
& \alpha_1,\alpha_2,\lambda_1,\lambda_2 \in \mathbb{Z}_+ && (6)\\
& \theta_{1i} \in \{0,1\} \quad \forall i && (7)\\
& \theta_{2j} \in \{0,1\} \quad \forall j && (8)
\end{aligned}
$$

Second Stage Variables

$$\phi(\alpha,\lambda,k) = \min \sum_t C^\beta(\beta_{1t}^k + \beta_{2t}^k) + \sum_{pt} C_p^z z_{pt}^k + \sum_{spt} C_{sp}^z z_{spt}^k + \sum_{spt} C_p^r r_{spt}^k \tag{9}$$

$$+ \sum_{vt} C_v^f n_{vt}^k + \sum_{st} C_s^f n_{st}^k + \sum_{vpt} C_v^v V_p x_{vpt}^k + \sum_{spt} C_s^v V_p y_{spt}^k$$

s.t.

$$\sum_{vp} x_{vpt}^k \le \lambda_1(\alpha_1 + \eta_1 \beta_{1t}^k) \qquad \forall t \tag{10}$$

$$\sum_{sp} y_{spt}^k \le \lambda_2(\alpha_2 + \eta_2 \beta_{2t}^k) \qquad \forall t \tag{11}$$

$$\beta_{2t}^k \le \delta\alpha_2;\ \beta_{1t}^k \le \delta\alpha_1 \qquad \forall t \tag{12}$$

$$z_{pt}^k - z_{p(t-1)}^k = \sum_v x_{vpt}^k - \sum_s y_{spt}^k \qquad \forall p, t \tag{13}$$

$$z_{spt}^k - z_{sp(t-1)}^k = \left[y_{spt}^k + r_{spt}^k\right] - \left[d_{spt}^k + r_{sp(t-1)}^k\right] \qquad \forall s, p, t \tag{14}$$

$$\sum_p V_p x_{vpt}^k \le Q n_{vt}^k \qquad \forall v, t \tag{15}$$

$$\sum_p V_p y_{spt}^k \le Q n_{st}^k \qquad \forall s, t \tag{16}$$

$$\beta_{1t}^k, \beta_{2t}^k, x_{vpt}^k, z_{pt}^k, y_{spt}^k, r_{spt}^k, n_{vt}^k, n_{st}^k \in \mathbb{Z}_+ \qquad \forall s, v, p, t \tag{17}$$

Linearization of the Bilinear Constraints

Reformulation-Linearization Technique (RLT)

Suppose that α is a continuous variable in the range $[0, M]$ and that θ is a continuous variable in the range $[0, 1]$. Then the following four inequalities can be written based on that knowledge of the bounds of the variables:

$$\begin{aligned}
(\theta - 0)(\alpha - 0) &\geq 0 \\
(\theta - 0)(\alpha - M) &\leq 0 \\
(\theta - 1)(\alpha - 0) &\leq 0 \\
(\theta - 1)(\alpha - M) &\geq 0
\end{aligned}
\quad \xrightarrow{\zeta \;=\; \alpha\theta} \quad
\begin{aligned}
0 \leq \quad \zeta \quad &\leq M\theta \\
0 \leq \alpha - \zeta &\leq M(1 - \theta)
\end{aligned}$$

Changes in Formulation of SWITP using RLT

First-Stage

$$\gamma A_1 \leq \sum_i \Lambda_{1i}\zeta_{1i} \qquad (4)$$

$$\gamma A_2 \leq \sum_j \Lambda_{2j}\zeta_{2j} \qquad (5)$$

$$0 \leq \zeta_i \leq M_\alpha \theta_{1i} \qquad \forall i \qquad (*)$$

$$0 \leq \alpha_1 - \zeta_i \leq M_\alpha (1 - \theta_{1i}) \qquad \forall i \qquad (*)$$

$$0 \leq \zeta_j \leq M_\alpha \theta_{2j} \qquad \forall j \qquad (*)$$

$$0 \leq \alpha_2 - \zeta_j \leq M_\alpha (1 - \theta_{2j}) \qquad \forall j \qquad (*)$$

$$\alpha_1, \alpha_2, \cancel{\lambda_1, \lambda_2} \in \mathbb{Z}_+ \qquad (6)$$

$$\theta_{1i}, \zeta_{1i} \in \{0, 1\} \qquad \forall i \qquad (7)$$

$$\theta_{2j}, \zeta_{2j} \in \{0, 1\} \qquad \forall j \qquad (8)$$

Changes in Formulation of SWITP using RLT
Second-Stage

$$\sum_{vp} x_{vpt}^k \le \sum_i \Lambda_{1i}(\zeta_{1i} + \eta_1 \xi_{1it}^k) \qquad \forall t \qquad (10)$$

$$\sum_{sp} y_{spt}^k \le \sum_j \Lambda_{2j}(\zeta_{2j} + \eta_2 \xi_{2jt}^k) \qquad \forall t \qquad (11)$$

$$0 \le \xi_{1it}^k \le M_\beta \theta_{1i} \qquad \forall i,t \qquad (*)$$

$$0 \le \beta_{1t}^k - \xi_{1it}^k \le M_\beta (1-\theta_{1i}) \qquad \forall i,t \qquad (*)$$

$$0 \le \xi_{2jt}^k \le M_\beta \theta_{2j} \qquad \forall j,t \qquad (*)$$

$$0 \le \beta_{2t}^k - \xi_{2jt}^k \le M_\beta (1-\theta_{2j}) \qquad \forall j,t \qquad (*)$$

$$\xi_{1it}^k \in \{0,1\} \qquad \forall i,t \qquad (*)$$

$$\xi_{2jt}^k \in \{0,1\} \qquad \forall j,t \qquad (*)$$

PURDUE ENGINEERING

Linearization of the Bilinear Constraints

Generalized Disjunctive Programming (GDP)

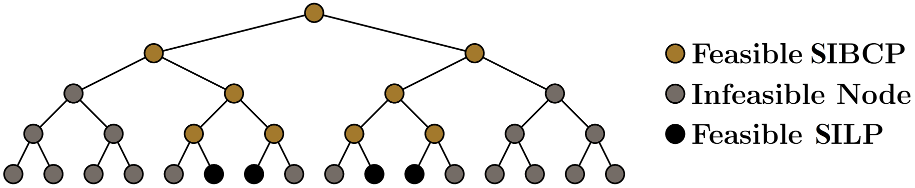
Disjunctive Programming

Disjunctive programming is linear programming with disjunctive constraints like the one below:

$$\bigvee_{j \in J_k} \begin{bmatrix} Y_{jk} \\ g_{jk}(x) \leq 0 \end{bmatrix}, \quad \forall k \in K$$

Linearization of the Bilinear Constraints

GDP Branch-and-Bound (GBB)



Implicit Enumeration

The bilinear terms results from the variable choice of technology. Once the choice of technology is fixed, the technology variables become technology parameters and the problem becomes linearly constrained. The problem resulting $I \times J$ stochastic integer linear problems can be solved in a branch-and-bound like framework.

Changes in Formulation of SWITP using GBB

$$\textit{Minimize} \quad C^{\alpha}(\alpha_1+\alpha_2)+C_{\theta_1}+C_{\theta_2}+\sum_k p_k\phi(\alpha,k) \tag{1}$$

$$\cancel{\textstyle\sum_i \theta_{1i}=1;\sum_j \theta_{2j}=1} \tag{2}$$

$$\cancel{\textstyle\sum_i \Lambda_{1i}\theta_{1i}=\lambda_1;\sum_j \Lambda_{2j}\theta_{2j}=\lambda_2} \tag{3}$$

$$\gamma A_1 \le \alpha_1\Lambda_1 \tag{4}$$

$$\gamma A_2 \le \alpha_2\Lambda_2 \tag{5}$$

$$\alpha_1,\alpha_2,\cancel{\lambda_1,\lambda_2}\in\mathbb{Z}_+ \tag{6}$$

$$\sum_{vp} x^k_{vpt} \le \Lambda_1(\alpha_1+\eta_1\beta^k_{1t}) \qquad \forall t \tag{10}$$

$$\sum_{sp} y^k_{spt} \le \Lambda_2(\alpha_2+\eta_2\beta^k_{2t}) \qquad \forall t \tag{11}$$

Linearization of the Bilinear Constraints

Big M

$$\bigvee_{j \in J_k} \begin{bmatrix} Y_{jk} \\ g_{jk}(x) \leq 0 \end{bmatrix}, \quad \forall k \in K$$

$$\Downarrow$$

$$\sum_j Y_{jk} = 1 \qquad \forall k \in K$$

$$g_{jk}(x) \leq M(1 - Y_{jk}) \qquad \begin{matrix} \forall k \in K \\ \forall j \in J_k \end{matrix}$$

Changes in Formulation of SWITP using Big-M

$$\bigvee_i \begin{bmatrix} \theta_i \\ \gamma A_1 \leq \alpha_1 \Lambda_{1i} \\ \sum_{vp} x^k_{vpt} \leq \Lambda_{1i}(\alpha_1 + \eta_1 \beta^k_{1t}) \end{bmatrix} ; \bigvee_j \begin{bmatrix} \theta_j \\ \gamma A_2 \leq \alpha_2 \Lambda_{2j} \\ \sum_{sp} y^k_{spt} \leq \Lambda_{2j}(\alpha_2 + \eta_2 \beta^k_{2t}) \end{bmatrix} \quad \forall t \quad \begin{matrix} (2) \\ (4),(5) \\ (10),(11) \end{matrix}$$

$$\Downarrow$$

$$\gamma A_1 \theta_{1i} \leq \alpha_1 \Lambda_{1i} \qquad \forall i \qquad (4)$$

$$\gamma A_2 \theta_{2j} \leq \alpha_2 \Lambda_{2j} \qquad \forall j \qquad (5)$$

$$\alpha_1, \alpha_2, \cancel{\lambda_1, \lambda_2} \in \mathbb{Z}_+ \qquad (6)$$

$$\sum_{vp} x^k_{vpt} - \Lambda_{1i}(\alpha_1 + \eta_1 \beta^k_{1t}) \leq M_x(1 - \theta_{1i}) \qquad \forall i, t \qquad (10)$$

$$\sum_{sp} y^k_{spt} - \Lambda_{2j}(\alpha_2 + \eta_2 \beta^k_{2t}) \leq M_y(1 - \theta_{2j}) \qquad \forall j, t \qquad (11)$$

PURDUE ENGINEERING

Generic Scenario Decomposition

Verbal Explaination

- Step 0 Introduce copies of the first-stage variables x for each scenario k in the set K $[x^1, x^2, \ldots, x^K]$ and add the non-anticipativity constraint $x^1 = x^2 = \cdots = x^K$.
- Step 1 Relax the non-anticipativity constraint so that you have K independent, but very similar, problems.
- Step 2 For each scenario k, solve the corresponding subproblem, subject to the problem constraints, to optimality (or at least to a feasible solution).
- Step 3 Obtain $\bar{x}$, determined by some predetermined method of averaging the values of x^k to come up with some concensus value.
- Step 4 For each scenario k, solve the corresponding subproblem, subject to the problem constraints, plue some term(s) that penalizes the lack of adherence to the non-anticipativity constraint to optimality (or at least to a feasible solution).
- Step 5 Determine an upper and lower bound to the problem.
- Step 6 If the gap is larger than desired and the alloted computer time has not elapsed, go to Step 3.
- Step 7 Post-process, if needed, to get a fully admissible and implementable solution.

Generic Scenario Decomposition

Mathematical Explanation

$$\min_{x,\ y_k} \{c^\top x + \sum_{k=1}^{K} p_k q_k^\top y_k : (x, y_k) \in S_k \subseteq \mathbb{Z}_+\} \qquad \text{(EF)}$$

Consider the following two-stage stochastic integer program (EF). Through the introduction of copies of the first-stage variables $x^1, x^2, \ldots, x^K$ [Step 0] we can rewrite it into K subproblems [Step 1].

$$\min_{x_k,\ y_k} \{\sum_{k=1}^{K} z_k(x_k, y_k) : (x_k, y_k) \in S_k, x^1 = x^2 = \cdots = x^K\} \qquad \text{(DD)}$$

$$z_k(x_k, y_k) = p_k(c^\top x_k + q_k^\top y_k)$$

The penalty [Step 4] is enforced by adding a term $\mu \sum_{k=1}^{K} H_k x_k$ such that the subproblems are still separable and $H_k x_k = 0$ when non-anticipativity is realized. This is referred to a Lagrangian relaxation. This new augmented problem is stated below.

$$D(\mu, \bar{x}) = \sum_{k=1}^{K} \min_{x_k,\ y_k} \{z_k(x_k, y_k) + \mu H_k x_k : (x_k, y_k) \in S_k\}$$

Generic Scenario Decomposition

Mathematical Explaination [cont'd]

For the Progressive Hedging Algorithm this penalty term is given:

$$\mu H_k x_k = w_k x_k + \frac{\rho}{2}\|x_k - \bar{x}\|^2$$

where ρ is a tuning parameter and w takes the place of μ and is updated

$$w_k^{(n)} = w_k^{(n-1)} + \rho\left(x_k^{(n-1)} - \bar{x}^{(n-1)}\right)$$

For the Augmented Lagrangian Dual Algorithm this penalty term is given:

$$\mu H_k x_k = \mu\|x_k - \bar{x}\|^2$$

where μ is updated

$$\mu^{(n)} = \mu^{(n-1)} + \frac{\alpha\left(UB^{(n-1)} - LB^{(n-1)}\right)}{\left(x_k^{(n-1)} - \bar{x}^{(n-1)}\right)^2} = f\left(\mu^{(n-1)}\right)$$

Scenario Decomposition Algorithm

$n \leftarrow 0;$

$\mu^{(0)} \leftarrow \mathbf{0}$ **OR** $w^{(0)} \leftarrow \mathbf{0};$

$LB^{(0)} \leftarrow D(\mu^{(0)}, \mathbf{0});$

$\bar{x}^{(0)} \leftarrow \sum_{k=1}^{K} p_k x_k^{(0)};$

$UB^{(0)} \leftarrow \sum_{k=1}^{K} z_k(\bar{x}^{(0)}, y_k);$

$\delta^{(0)} \leftarrow 1 - \frac{LB^{(0)}}{UB^{(0)}};$

while $\underline{\delta^{(n)} \geq .0001}$ **do**

    $n \leftarrow n + 1;$

    $\mu^{(n)} \leftarrow f(\mu^{(n-1)});$

    $LB^{(n)} \leftarrow D(\mu^{(n)}, \bar{x}^{(n-1)});$

    $\bar{x}^{(n)} \leftarrow \sum_{k=1}^{K} p_k x_k^{(n)};$

    $UB^{(n)} \leftarrow \sum_{k=1}^{K} z_k(\bar{x}^{(n)}, y_k)$;

    $\delta^{(n)} \leftarrow 1 - \frac{LB^{(n)}}{UB^{(n)}};$

end

[Step 0] - [Step 1] done before

[Step 2]

[Step 3]

[Step 4]

[Step 5]

[Step 6]

[Step 7] not shown

Instance Generation

- The parameters are generated by a Python script dynamically and randomly based on the desired number of stores, vendors, products, tech choices, time periods, and scenarios.
- The demand is a discrete uniform distribution between 75 – 125% the value from the previous time period for a given store and product.
- The technology rates and cost are generated from a uniform distribution and then rank sorted as to make the rate and cost perfectly correlated.
- The parameters are then written to .dat files that are the input to the Abstract model created in Pyomo.

Preliminary Results

Average CPU Seconds

$I \times J$	K	GBB	RLT	GBB	BigM
		EF	PH		
3×3	5	63	3	21	3
5×5	5	145	59	54	33
10×10	5	684	143	225	69
15×15	5	1564	191	538	129

$I \times J$	K	GBB	RLT	GBB	BigM
		EF	PH		
3×3	20	-	123	73	128
5×5	20	-	424	231	333
10×10	20	-	1118	866	403
15×15	20	-	1285	1912	484

$I \times J$	K	GBB	RLT	GBB	BigM
		EF	PH		
3×3	10	-	6	39	5
5×5	10	-	127	185	68
10×10	10	-	406	628	228
15×15	10	-	423	1010	253

$I \times J$	K	GBB	RLT	GBB	BigM
		EF	PH		
3×3	50	-	269	175	22
5×5	50	-	762	511	721
10×10	50	-	1158	2012	766
15×15	50	-	1367	4669	1064

Modeling Language: Coopr 3.5.8787

Optimation Solver: PySP using Gurobi 5.63

PURDUE ENGINEERING

Preliminary Results

MINIMUM CPU Seconds

$I \times J$	K	GBB	RLT	GBB	BigM
		EF	PH		
3×3	5	48	3	18	2
5×5	5	126	3	49	2
10×10	5	684	108	199	3
15×15	5	1564	156	486	110

$I \times J$	K	GBB	RLT	GBB	BigM
		EF	PH		
3×3	20	-	10	68	9
5×5	20	-	15	223	12
10×10	20	-	679	774	329
15×15	20	-	619	1731	386

$I \times J$	K	GBB	RLT	GBB	BigM
		EF	PH		
3×3	10	-	5	36	4
5×5	10	-	6	113	5
10×10	10	-	242	376	5
15×15	10	-	324	943	213

$I \times J$	K	GBB	RLT	GBB	BigM
		EF	PH		
3×3	50	-	26	158	21
5×5	50	-	41	464	25
10×10	50	-	1006	1855	24
15×15	50	-	1287	4183	847

Modeling Language: Coopr 3.5.8787

Optimation Solver: PySP using Gurobi 5.63

Preliminary Results
Comparison

	Wins	Speedup
RLT	1	x1.7
GBB	12	-
Big M	83	x2.8

Conclusion

- Bilinear constraints can be handled in different ways.
- The big-M reformulation is, on average, about 2.8 times faster than the GDP Branch-and-Bound (GBB) in the preliminary result.
- The GBB is more robust with significantly smaller standard deviations on the runs.

References

[1] S. Lee and I. E. Grossmann, New algorithms for nonlinear generalized disjunctive programming, Comput. Chem. Eng., vol. 24, no. 910, pp. 21252141, Oct. 2000.

[2] W. P. Adams and H. D. Sherali, Mixed-integer bilinear programming problems, Math. Program., vol. 59, pp. 279305, 1993.

[3] G. L. Nemhauser, L. A.Wolsey, Integer and Combinatorial Optimization,Wiley-Interscience, New York 1988.

[4] F. Trespalacios and I. E. Grossmann, Review of Mixed-Integer Nonlinear and Generalized Disjunctive Programming Methods, Chemie Ing. Tech., vol. 86, no. 7, pp. 9911012, Jul. 2014.

Questions?

Thank You!