Optimizing Supply Blocks for Expeditionary Units

Jefferson Huang

Assistant Professor
Operations Research Department
Naval Postgraduate School

The 90th MORS Symposium

Marine Corps University, Quantico, VA

14 June, 2022

Marine Expeditionary Units (MEUs)

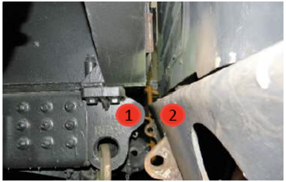
- Compact Marine air-ground task forces (MAGTFs) capable of rapidly responding to crisis situations
- 3 HQ'd at Camp Pendleton, 3 HQ'd at Camp Lejeune, 1 permanently forward-deployed MEU HQ'd in Okinawa
- Operate according to 15-month deployment cycle

  Deployments typically include periods (e.g., weeks) during which **external resupply is infeasible**.

Supply Blocks for MEUs

- MEUs typically deploy with **blocks** of materiel; e.g., Class IX (repair) parts
- Currently developed by Supply Management Units (SMUs) at Marine Corps Logistics Command (MARCORLOGCOM)
- Intended to provide roughly 15 days of supplies for deployed MEUs.

Huge number of potentially combat-essential parts, limited storage capacity!



The Problem, and Our Contributions

The Problem: How many units of each combat-essential part should a given MEU bring with it on its next 6-month deployment, given physical constraints on how much can be taken?

Importance: The deployed MEU will have limited opportunities for timely external resupply.

Solutions Without Our Help: Current MS VBA-based methods aim to minimize the expected number of shortages, subject to a single (e.g. volume) constraint. **Parts are selected using various heuristics**.

- e.g., About 6 hours to consider roughly 3,200 National Item Identification Numbers (NIINs), or about 30 minutes with a highly skilled operator.

What Did We Do?

- Formulated an **optimization model** that includes all known prior formulations as special cases.
  - Accounts for things existing methods do not (e.g., penalties for left-over parts, multiple constraints).
- Formulated an equivalent *linear* optimization model that is efficiently solvable (e.g., roughly 5 minutes for a realistically-sized instance derived from 2018 deployment data from the 18th MEU).

INCREASING EFFICIENCY OF CLASS IX REPAIR
PART BLOCKS FOR DEPLOYED UNITS

by

Nikolas C. Anthony

June 2021

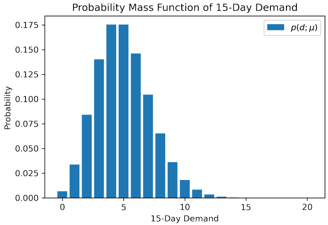
Thesis Advisor:	Jefferson Huang
Second Reader:	Peter A. Nesbitt

Demand Assumptions

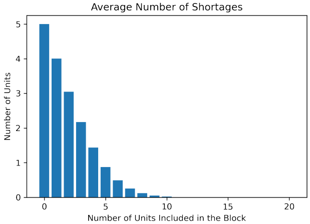
The demand D_i for each NIIN is modeled as an independent Poisson random variable:

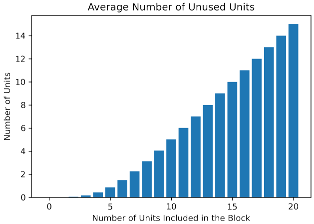
$$P(D_i = d) = \frac{\mu^d}{d!} \cdot e^{-\mu} =: p(d; \mu)$$

- Completely specified by the **mean demand** μ during the time interval of interest (e.g., 15 days)
- **Example:** Demand distribution for a part with an average of $\mu = 5$ units of demand during 15 days.



Objective: Balance shortages with excess units, subject to physical constraints.





The (Nonlinear) Optimization Model

Input Data:

- I = total number of item types (e.g., NIINs)
- J = total number of constraints (e.g., volume, weight)
- ℓ_i = cost per un-used unit of item i
- b_i = cost per unit short for item i
- μ_i = mean demand for item i during 15 days
- $v_{i,j}$ = units of constraint j consumed by item i
- V_j = total units of constraint j available

Decision Variables:

- s_i = number of units of item i to include in the supply block

Average cost of stocking s_i units of item i:

$$G_i(s_i) = \ell_i \sum_{d=0}^{s_i} (s_i - d)p(d;\mu_i) + b_i \sum_{d=s_i+1}^{\infty} (d - s_i)p(d;\mu_i)$$

Nonlinear Optimization Model:

$$\begin{array}{lll} \text{minimize} & \displaystyle\sum_{i=1}^{I} G_i(s_i) & \\ \text{subject to} & \displaystyle\sum_{i=1}^{I} v_{i,j} s_i \le V_j & j = 1, \ldots, J \\ & s_i \in \{0, 1, \ldots\}, & i = 1, \ldots, I \end{array}$$

The Equivalent Linear Optimization Model

Idea: Linearize the objective function:

1. Introduce auxiliary decision variables $y_{i,k}$ for each item i and stock level k, so that

$$s_i = \sum_{i=1}^{K} y_{i,k}, \quad \text{for each item } i,$$

where K is the maximum stock level of item i.

2. Introduce auxiliary cost coefficients $\delta_{i,k}$ for each i and k, where

$$\delta_{i,k} = -P(D_i > k)$$

Theorem: The Nonlinear Optimization Model is equivalent to the following **linear** optimization model:

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^{I} \left[\ell_i \left(\sum_{k=0}^{K} y_{i,k} - \mu_i \right) + (\ell_i + b_i) \left(\mu_i + \sum_{k=0}^{K} \delta_{i,k} y_{i,k} \right) \right] \\ \text{subject to} & \displaystyle\sum_{i=1}^{I} v_{i,j} \left(\sum_{k=0}^{K} y_{i,k} \right) \leq V_j, \qquad j = 1, \ldots, J \\ & y_{i,k} \in \{0, 1\}, \qquad k = 1, \ldots, K, \ \ i = 1, \ldots, I \end{array}$$

Straightforward to implement in, e.g., Pyomo.

How Does It Do?

Implemented as "OptiStock" in Pyomo by Capt Nikolas Anthony (NPS MS in OR June 2021, currently at MARCORLOGCOM)

Experimental model parameters used to compute stocking policies:

	Leftover Costs					Backorder Costs					Max Units
Model Name	**A**	**B**	**C**	**D**	**E**	**A**	**B**	**C**	**D**	**E**	
OptiStock 1	1	1	1	1	1	1	1	1	1	1	70
OptiStock 2	0	0	0	0	0	1	1	1	1	1	70
ABC OptiStock 1	0.2	0.4	0.6	0.8	1	1	0.8	0.6	0.4	0.2	70
ABC OptiStock 2	0	0	0	0	0	1	0.8	0.6	0.4	0.2	70
ABC OptiStock 3	0.2	0.4	0.6	0.8	1	1	0.8	0.6	0.4	0.2	1000
ABC OptiStock 4	0	0	0	0	0	1	0.8	0.6	0.4	0.2	1000
ABC OptiStock 5	0	0	0	0	0	1	0.5	0.25	0.125	0.0625	1000

Computed stocking policies compared with existing "Block Builder" stocking policy on 2018 data from 13th MEU deployment.

How Does It Do?

Results: Reduced costs, comparable/better performance in terms of number of shortages and leftovers, significantly reduced run time (from hours, to minutes on a student laptop).

- Student thesis is CUI, available upon request as appropriate.

What's Next?

- More extensive numerical experiments?
- More efficient implementations?
- Incorporate additional information (e.g., DLA posture information, results from simulations)?
- Engage with MARCORLOGCOM to positively influence the block building process.